aws

Guía del usuario

Amazon Simple Storage Service

Versión de API 2006-03-01

Amazon Simple Storage Service: Guía del usuario

Table of Contents

¿Qué es Amazon S3?

Amazon Simple Storage Service (Amazon S3) es un servicio de almacenamiento de objetos que ofrece escalabilidad, disponibilidad de datos, seguridad y rendimiento líderes del sector. Los clientes de todos los tamaños y sectores pueden utilizar Amazon S3 para almacenar y proteger cualquier cantidad de datos para diversos casos de uso, tales como lagos de datos, sitios web, aplicaciones móviles, copia de seguridad y restauración, archivado, aplicaciones empresariales, dispositivos IoT y análisis de big data. Amazon S3 proporciona funciones de gestión para que pueda optimizar, organizar y configurar el acceso a sus datos para satisfacer sus requisitos empresariales, organizativos y de conformidad específicos.

> **Note**
>
> Para obtener más información sobre el uso de la clase de almacenamiento Amazon S3 Express One Zone con buckets de directorio, consulte ¿Qué es S3 Express One Zone? y Buckets de directorio.

Temas

Características de Amazon S3

Clases de almacenamiento

Amazon S3 ofrece varios tipos de almacenamiento diseñados para distintos casos de uso. Por ejemplo, puede almacenar datos de producción críticos en S3 Standard o S3 Express One Zone para obtener acceso frecuente, ahorrar costes al almacenar datos a los que se accede con poca

frecuencia en S3 Standard-IA o S3 One Zone-IA, y archivar datos con los costos más bajos en S3 Glacier Instant Retrieval, S3 Glacier Flexible Retrieval y S3 Glacier Deep Archive.

Amazon S3 Express One Zone es una clase de almacenamiento de Amazon S3 en zona única de alto rendimiento que está diseñada específicamente para ofrecer acceso constante a los datos en milisegundos de un solo dígito para los datos a los que accede para las aplicaciones sensibles a la latencia. S3 Express One Zone es la clase de almacenamiento de objetos en la nube con la latencia más baja disponible en la actualidad, con una velocidad de acceso a los datos hasta 10 veces más rápida y unos costos de solicitud un 50 % más bajos que los de S3 Standard. S3 Express One Zone es la primera clase de almacenamiento de S3 en la que se puede seleccionar una única zona de disponibilidad con la opción de coubicar su almacenamiento de objetos junto con sus recursos informáticos, lo que brinda la mayor velocidad de acceso posible. Además, para aumentar aún más la velocidad de acceso y admitir cientos de miles de solicitudes por segundo, los datos se almacenan en un nuevo tipo de bucket: un bucket de directorio de Amazon S3. Para obtener más información, consulte ¿Qué es S3 Express One Zone? y Buckets de directorio.

Puede almacenar datos con patrones de acceso cambiantes o desconocidos en S3 Intelligent-Tiering, que optimiza los costos de almacenamiento moviendo automáticamente los datos entre cuatro niveles de acceso cuando cambian los patrones de acceso. Funciona con el almacenamiento de objetos en cuatro capas de acceso: dos de acceso de baja latencia optimizadas para el acceso frecuente y poco frecuente y dos de acceso a archivos opcionales diseñados para el acceso asíncrono, las cuales están optimizadas para accesos inusuales.

Para obtener más información, consulte Uso de las clases de almacenamiento de Amazon S3.

Administrar el almacenamiento

Amazon S3 cuenta con funciones de gestión del almacenamiento que puede utilizar para gestionar los costes, cumplir los requisitos normativos, reducir la latencia y guardar varias copias distintas de sus datos para cumplir los requisitos de cumplimiento.

- Ciclo de vida de S3: defina la configuración de ciclo de vida para administrar los objetos y almacenarlos de manera económica durante todo su ciclo de vida. Puede realizar la transición de objetos a otras clases de almacenamiento de S3 o caducar objetos que alcancen el final de su vida útil.
- S3 Object Lock: evite que se eliminen o se sobreescriban objetos de Amazon S3 durante un período de tiempo determinado o de manera indefinida. Object Lock se puede utilizar para cumplir con los requisitos normativos que requieren almacenamiento de escritura única y lectura múltiple

(WORM) o simplemente para agregar otra capa de protección para evitar cambios y eliminaciones de objetos.

- Replicación de S3— Replique objetos y sus respectivos metadatos y etiquetas de objeto en uno o más buckets de destino en el mismo o en diferentesRegiones de AWSpara reducir la latencia, el cumplimiento normativo, la seguridad y otros casos de uso.
- Operaciones por lotes de S3: gestione miles de millones de objetos a escala con una sola solicitud de API de S3 o con unos pocos clics en la consola de Amazon S3. Puede utilizar Operaciones Batch para realizar operaciones comoCopy (Copiar),InvocaciónAWSFunción de Lambda, yRestauraren millones o miles de millones de objetos.

Gestión de acceso y seguridad

Amazon S3 proporciona funciones para auditar y gestionar el acceso a sus buckets y objetos. De forma predeterminada, los buckets y los objetos de S3 son privados. Solo tiene acceso a los recursos de S3 que cree. Para conceder permisos de recursos detallados que admitan su caso de uso específico o para auditar los permisos de sus recursos de Amazon S3, puede utilizar las siguientes características.

- S3 Block Public Access: bloquea el acceso público a los buckets y objetos de S3. De forma predeterminada, la configuración de bloqueo del acceso público se activa en el nivel de bucket. Le recomendamos que deje todas las configuraciones habilitadas a menos que sepa que necesita desactivar una o varias para su caso de uso concreto. Para obtener más información, consulte Establecer la configuración de Block Public Access para sus buckets de S3.
- AWS Identity and Access Management (IAM): IAM es un servicio web que le ayuda a controlar de forma segura el acceso a los recursos de AWS, como los recursos de Amazon S3. Con IAM, se pueden administrar de forma centralizada los permisos que controlan a qué recursos de AWS pueden acceder los usuarios. Utilice IAM para controlar quién está autenticado (ha iniciado sesión) y autorizado (tiene permisos) para utilizar recursos.
- Políticas de buckets: utilice el lenguaje de políticas basado en IAM para configurar permisos basados en recursos para los buckets de S3 y los objetos que hay en ellos.
- Puntos de acceso de Amazon S3: configure los puntos de acceso de la red con nombre con políticas de acceso dedicadas para administrar el acceso a los datos a escala para los conjuntos de datos compartidos en Amazon S3.
- Listas de control de acceso (ACL): conceder permisos de lectura y escritura para buckets y objetos individuales a usuarios autorizados. Como regla general, se recomienda utilizar políticas basadas

en recursos de S3 (políticas de bucket y políticas de punto de acceso) o políticas de usuario de IAM para el control de acceso en lugar de las ACL. Las políticas son una opción de control de acceso simplificada y más flexible. Con las políticas de bucket y las políticas de puntos de acceso, puede definir reglas que se apliquen ampliamente a todas las solicitudes a sus recursos de Amazon S3. Para obtener más información acerca de casos específicos en que usaría ACL en lugar de políticas basadas en recursos o políticas de usuarios de IAM, consulte Administración de acceso con ACL.

- S3 Object Ownership: tome posesión de cada objeto del bucket, lo que simplificará la administración del acceso a los datos almacenados en Amazon S3. S3 Object Ownership es una configuración en el nivel de bucket de Amazon S3 que puede usar para desactivar o activar las ACL. Las ACL están desactivadas de forma predeterminada. Cuando las ACL están desactivadas, el propietario del bucket posee todos los objetos del bucket y administra el acceso a los datos de forma exclusiva mediante políticas de administración de acceso.
- Analizador de acceso de IAM para S3: evalúe y monitoree sus políticas de acceso al bucket de S3, asegurándose de que las políticas solo proporcionen el acceso previsto a sus recursos de S3.

Procesamiento de datos

Para transformar datos y activar flujos de trabajo para automatizar una variedad de otras actividades de procesamiento a escala, puede utilizar las siguientes características.

- S3 Object Lambda: agregue su propio código a las solicitudes GET, HEAD y LIST de S3 para modificar y procesar los datos a medida que vuelven a una aplicación. Filtra filas, redimensiona dinámicamente las imágenes, redacta datos confidenciales y mucho más.
- Notificaciones de eventos: active flujos de trabajo que utilizan Amazon Simple Notification Service (Amazon SNS), Amazon Simple Queue Service (Amazon SQS) yAWS Lambdacuando se realiza un cambio en los recursos de S3.

Registro y monitorización

Amazon S3 proporciona herramientas de registro y supervisión que puede utilizar para supervisar y controlar cómo se utilizan sus recursos de Amazon S3. Para obtener más información sobre la monitorización de , consulte .

Herramientas de monitoreo automatizadas

- Métricas de Amazon CloudWatch para Amazon S3: realice un seguimiento del estado operativo de sus recursos de S3 y configure alertas de facturación cuando los cargos estimados alcancen un umbral definido por el usuario.
- AWS CloudTrail proporciona un registro de las medidas adoptadas por un usuario, un rol o un servicio de Servicio de AWS en Amazon S3. Los registros de CloudTrail le proporcionan un seguimiento detallado de la API para las operaciones a nivel de bucket y de objeto de Amazon S3.

Herramientas de monitoreo manuales

- Registro de acceso al servidor: brinda registros detallados para las solicitudes realizadas a un bucket. Puede utilizar los registros de acceso al servidor para auditorías de seguridad y acceso, obtener información sobre la base de clientes o comprender la factura de Amazon S3.
- AWSTrusted Advisor— Evalúe su cuenta usandoAWScomprobaciones de prácticas recomendadas para identificar formas de optimizar suAWSinfraestructura, mejorar la seguridad y el rendimiento, reducir los costos y supervisar las cuotas de servicio. A continuación, puede seguir las recomendaciones para optimizar sus servicios y recursos.

Uso de análisis e información

Amazon S3 ofrece funciones que le ayudarán a obtener visibilidad del uso del almacenamiento, lo que le permite comprender mejor, analizar y optimizar su almacenamiento a escala.

- Amazon S3 Storage Lens: comprenda, analice y optimice su almacenamiento. S3 Storage Lens proporciona más de 60 métricas de uso y actividad y paneles interactivos para añadir datos para toda su organización, cuentas específicas, Regiones de AWS, buckets o prefijos.
- Análisis de clases de almacenamiento: analice los patrones de acceso al almacenamiento para decidir cuándo es el momento de mover los datos a una clase de almacenamiento más rentable.
- Informes de inventario de S3 con informes de inventario: auditar e informar sobre los objetos y sus metadatos correspondientes y configurar otras funciones de Amazon S3 para tomar medidas en los informes de inventario. Por ejemplo, puede informar sobre el estado de replicación y cifrado de los objetos. Para obtener una lista de todos los metadatos disponibles para cada objeto en los informes de inventario, consulteLista de inventario de Amazon S3.

Consistencia sólida

Amazon S3 proporciona una sólida coherencia de lectura tras escritura para las operaciones PUT y DELETE de objetos del bucket de Amazon S3 en todas las Regiones de AWS. Esto se aplica tanto a las escrituras en objetos nuevos como a las solicitudes PUT que sobrescriben objetos existentes y las solicitudes DELETE. Además, las operaciones de lectura en Amazon S3 Select, las listas de control de acceso de Amazon S3, las etiquetas de objeto de Amazon S3 y los metadatos de objetos (p. ej., el objeto HEAD) son muy consistentes. Para obtener más información, consulte Modelo de consistencia de datos de Amazon S3.

Cómo funciona Amazon S3

Amazon S3 es un servicio de almacenamiento de objetos que almacena datos como objetos dentro de buckets. Un objeto es un archivo y cualquier metadato que describa ese archivo. Un bucket es un contenedor de objetos.

Para almacenar datos en Amazon S3, primero debe crear un bucket y especificar un nombre de bucket yRegión de AWS. A continuación, cargue datos a ese bucket como objetos en Amazon S3. Cada objeto tiene unclave(oNombre de clave), que es el identificador único del objeto dentro del bucket.

S3 proporciona funciones que puede configurar para admitir su caso de uso específico. Puede utilizar S3 Versioning para mantener varias versiones de un objeto en un bucket y restaurar objetos que se eliminan o sobrescriben accidentalmente.

Los buckets y los objetos que contienen son privados y solo se puede acceder a ellos si concede explícitamente permisos de acceso. Puede utilizar políticas de bucket,AWS Identity and Access Management(IAM), listas de control de acceso (ACL) y puntos de acceso S3 para administrar el acceso.

Temas

- Buckets
- Objects
- Claves
- Control de versiones de S3
- ID de versión.
- Política de bucket

- Puntos de acceso de S3
- Listas de control de acceso (ACL)
- Regiones

Buckets

Un bucket es un contenedor para objetos almacenados en Amazon S3. Puede almacenar cualquier cantidad de objetos en un bucket y puede tener hasta 100 buckets en su cuenta. Para solicitar un aumento, visite la Consola de Service Quotas.

Cada objeto está almacenado en un bucket. Por ejemplo, si el objeto denominado `photos/puppy.jpg` se almacena en el bucket `amzn-s3-demo-bucket` en la región Oeste de EE. UU. (Oregón), se puede redirigir con la URL `https://amzn-s3-demo-bucket.s3.us-west-2.amazonaws.com/photos/puppy.jpg`. Para obtener más información, consulte Acceso bucket.

Cuando crea un bucket, introduzca un nombre de bucket y elija la opción Región de AWS donde residirá el bucket. Después de crear un bucket, no se puede cambiar su nombre ni su región. Los nombres de los buckets deben seguir laReglas de nomenclatura bucket. También puede configurar un bucket para utilizarControl de versiones de S3u otrosAdministrar el almacenamientoCaracterísticas de.

Cucharones también:

- Organizan el espacio de nombres de Amazon S3 al más alto nivel.
- Identifican la cuenta responsable para los cargos de almacenamiento y transferencia de datos.
- Proporcione opciones de control de acceso, tales como políticas de bucket, listas de control de acceso (ACL) y puntos de acceso de S3, que puede utilizar para administrar el acceso a sus recursos de Amazon S3.
- Sirven como la unidad de agregación para informes de uso.

Para obtener más información acerca de los buckets, consulte Descripción general de los buckets.

Objects

Los objetos son las entidades fundamentales almacenadas en Amazon S3. Los objetos se componen de datos de objetos y metadatos. Los metadatos son conjuntos de pares nombre-valor

que describen el objeto. Incluyen algunos metadatos predeterminados, como la fecha de la última modificación y los metadatos HTTP estándar, como `Content-Type`. También puede especificar metadatos personalizados en el momento en que se almacena el objeto.

Un objeto se identifica de forma exclusiva dentro de un bucket con una clave (nombre) y un ID de versión (si el control de versiones de S3 está habilitado en el bucket). Para obtener más información sobre los objetos, consulte Información general de los objetos de Amazon S3.

Claves

Una clave de objeto (o nombre de clave) es el identificador único de un objeto dentro de un bucket. Cada objeto de un bucket tiene exactamente una clave. La combinación de un bucket, clave de objeto y, opcionalmente, el ID de versión (si el control de versiones de S3 está habilitado para el bucket) identifica de forma única cada objeto. Por tanto, puede pensar en Amazon S3 como una asignación de datos básica entre "bucket + clave + versión" y el objeto en sí.

Se puede acceder a cada objeto de Amazon S3 de forma exclusiva a través de la combinación de punto de conexión de servicio web, nombre del bucket, clave, y de forma opcional, una versión. Por ejemplo, en la URL `https://amzn-s3-demo-bucket.s3.us-west-2.amazonaws.com/photos/puppy.jpg`, *`amzn-s3-demo-bucket`* es el nombre del bucket y `photos/puppy.jpg` es la clave.

Para obtener más información sobre las claves de objetos, consulte Creación de nombres de clave de objeto.

Control de versiones de S3

Puede usar el control de versiones de S3 para conservar diversas variantes de un objeto en el mismo bucket. Puede utilizar el control de versiones de S3 para conservar, recuperar y restaurar todas las versiones de los objetos almacenados en su bucket de . Puede recuperarse fácilmente de acciones no deseadas del usuario y de errores de la aplicación.

Para obtener más información, consulte Usar el control de versiones en buckets de S3.

ID de versión.

Si activa el control de versiones de S3 en un bucket, Amazon S3 genera un ID de versión único para cada objeto agregado al bucket. Los objetos que ya existían en el bucket en el momento en que habilita el control de versiones tienen un ID de versión de `null`. Si modifica estos objetos (o

cualquier otro) con otras operaciones, como CopyObject y PutObject, los objetos nuevos obtienen un ID de versión único.

Para obtener más información, consulte Usar el control de versiones en buckets de S3.

Política de bucket

Una política de bucket es una política AWS Identity and Access Management basada en recursos (IAM) que puede utilizar para conceder permisos de acceso al bucket y a los objetos que contiene. Solo el propietario del bucket puede asociar una política a un bucket. Los permisos asociados a un bucket se aplican a todos los objetos del bucket que son propiedad de la cuenta de propietario del bucket. Las políticas de bucket tienen un límite de tamaño de 20 KB.

Las políticas de buckets utilizan el lenguaje de políticas de acceso basado en JSON que es estándar en AWS. Puede utilizar directivas de bucket para agregar o denegar permisos para los objetos de un bucket. Las políticas de bucket permiten o deniegan solicitudes basadas en los elementos de la política, incluidos el solicitante, las acciones de S3, los recursos y los aspectos o condiciones de la solicitud (por ejemplo: la dirección IP utilizada para realizar la solicitud). Por ejemplo, puede crear una política de bucket que otorgue permisos entre cuentas para cargar objetos en un bucket de S3 y, al mismo tiempo, garantizar que el propietario del bucket tenga el control total de los objetos cargados. Para obtener más información, consulte Ejemplos de políticas de bucket de Amazon S3.

En su política de bucket, puede utilizar caracteres comodín en nombres de recursos de Amazon (ARN) y otros valores para otorgar permisos a un subconjunto de objetos. Por ejemplo, puede controlar el acceso a grupos de objetos que empiezan por unprefijoo terminar con una extensión dada, como `.html`.

Puntos de acceso de S3

Los puntos de acceso de Amazon S3 se denominan endpoints de red con políticas de acceso dedicadas que describen cómo se puede acceder a los datos mediante ese endpoint. Los puntos de acceso están asociados a los buckets que se pueden utilizar para realizar operaciones con objetos de S3, como GetObject y PutObject. Los puntos de acceso simplifican la administración del acceso a los datos a escala para los conjuntos de datos compartidos en Amazon S3.

Cada punto de acceso tiene su propia política de puntos de acceso. También puede configurar los parámetros de Bloquear acceso público para cada punto de acceso. Puede configurar cualquier punto de acceso para que acepte solo las solicitudes procedentes de una nube virtual privada (VPC) con el fin de restringir el acceso a los datos de Amazon S3 a una red privada.

Para obtener más información, consulte Administración del acceso a datos con puntos de acceso de Amazon S3.

Listas de control de acceso (ACL)

Puede utilizar las ACL para conceder permisos de lectura y escritura para buckets y objetos individuales a usuarios autorizados. Cada bucket y objeto incluye una ACL como un subrecurso. La ACL define qué Cuentas de AWS o grupos cuentan con acceso y el tipo de acceso que tienen. Las ACL son un mecanismo de control de acceso anterior a IAM. Para obtener más información acerca de las ACL, consulte Información general de las Listas de control de acceso (ACL).

S3 Object Ownership es una configuración de bucket de Amazon S3 que puede usar para controlar la propiedad de los objetos que se cargan en el bucket y para activar o desactivar las ACL. De forma predeterminada, la propiedad de objetos se establece en la configuración impuesta por el propietario del bucket. Además, todas las ACL están deshabilitadas. Cuando las ACL están deshabilitadas, el propietario del bucket posee todos los objetos del bucket y administra su acceso de forma exclusiva mediante políticas de administración de acceso.

La mayoría de los casos de uso modernos de Amazon S3 ya no requieren el uso de ACL. Le recomendamos desactivar las ACL, excepto en circunstancias inusuales en las que necesite controlar el acceso a cada objeto de manera individual. Si las ACL están desactivadas, puede usar políticas para controlar el acceso a todos los objetos del bucket, independientemente de quién haya subido los objetos al bucket. Para obtener más información, consulte Control de la propiedad de los objetos y desactivación de las ACL del bucket.

Regiones

Puede elegir la Región de AWS geográfica donde Amazon S3 almacenará los buckets que usted cree. Puede elegir una región para optimizar la latencia, minimizar los costos o cumplir con requisitos legales. Los objetos almacenados en una Región de AWS nunca abandonan la región, a menos que se transfieran o repliquen expresamente a otra región. Por ejemplo, los objetos almacenados en la región UE (Irlanda) nunca salen de ella.

> **Note**
>
> Solo puede tener acceso a Amazon S3 y sus características en las Regiones de AWS que estén habilitadas para su cuenta. Para obtener más información acerca de cómo habilitar una

región para crear y administrar recursos de AWS, consulte Administración de Regiones de AWS en la Referencia general de AWS.

Para ver una lista de las regiones y los puntos de conexión de Amazon S3, consulte Regiones y puntos de conexión en la Referencia general de AWS.

Modelo de consistencia de datos de Amazon S3

Amazon S3 proporciona una sólida coherencia de lectura tras escritura para las operaciones PUT y DELETE de objetos del bucket de Amazon S3 en todas las Regiones de AWS. Esto se aplica tanto a las escrituras en objetos nuevos como a las solicitudes PUT que sobrescriben objetos existentes y las solicitudes DELETE. Además, las operaciones de lectura en Amazon S3 Select, las listas de control de acceso de Amazon S3, las etiquetas de objeto de Amazon S3 y los metadatos de objetos (p. ej., el objeto HEAD) son muy consistentes.

Las actualizaciones en una sola clave son atómicas. Por ejemplo, si aplica PUT a una clave existente de un hilo y realiza una operación GET en la misma clave desde un segundo hilo simultáneamente, obtendrá los datos antiguos o los datos nuevos, pero nunca datos parciales o dañados.

Amazon S3 consigue una alta disponibilidad mediante la reproducción de los datos de varios servidores ubicados en los centros de datos de AWS. Si una solicitud PUT se realiza correctamente, sus datos se almacenan de forma segura. Cualquier lectura (solicitud GET o LIST) que se inicie después de recibir una respuesta PUT exitosa devolverá los datos escritos por la solicitud PUT. A continuación se muestran algunos ejemplos de este comportamiento:

- Un proceso escribe un nuevo objeto en Amazon S3 y enumera inmediatamente claves dentro del bucket. El nuevo objeto aparece en la lista.
- Un proceso reemplaza un objeto existente e inmediatamente intenta leerlo. Amazon S3 devuelve los datos nuevos.
- Un proceso elimina un objeto existente e inmediatamente intenta leerlo. Amazon S3 no devuelve ningún dato ya que el objeto se ha eliminado.
- Un proceso elimina un objeto existente y enumera inmediatamente claves dentro del bucket. El objeto no aparece en la lista.

> **Note**
>
> - Amazon S3 no admite el bloqueo de objetos para escritores simultáneos. Si se realizan dos solicitudes PUT simultáneamente a la misma clave, prevalece la solicitud con la marca temporal más reciente. Si esto es un problema, debe crear un mecanismo de bloque de objeto en su aplicación.
> - Las actualizaciones se basan en claves. No se pueden realizar actualizaciones atómicas en las claves. Por ejemplo, no puede realizar la actualización de una clave que depende de la actualización de otra clave, a menos que diseñe esta funcionalidad en su aplicación.

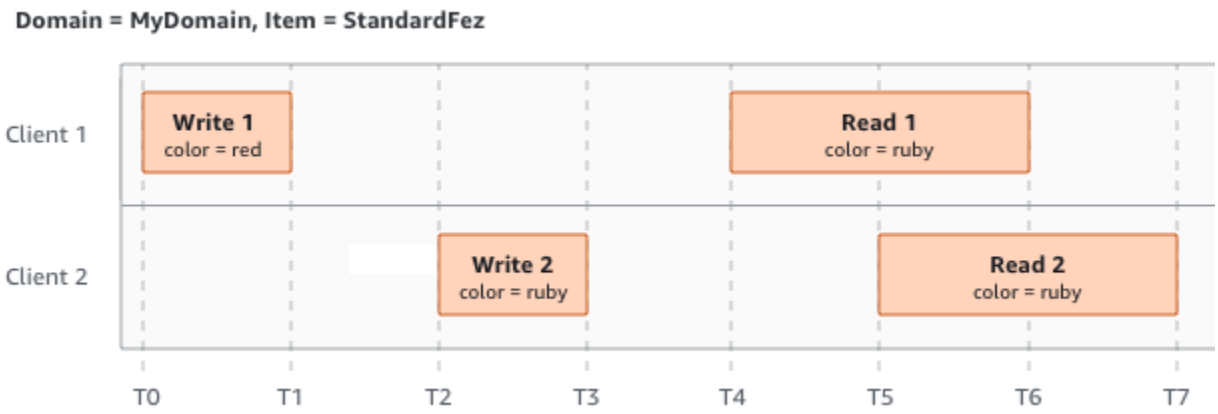
Las configuraciones del bucket tienen un modelo de consistencia final. En concreto, esto significa que:

- Si elimina un bucket e inmediatamente muestra todos los buckets, es posible que el bucket eliminado aún aparezca en la lista.
- Si habilita el control de versiones en un bucket por primera vez, es posible que el cambio se propague por completo en un instante. Para emitir operaciones de escritura (solicitudes PUT o DELETE) en los objetos del bucket, se recomienda que espere 15 minutos después de habilitar el control de versiones.

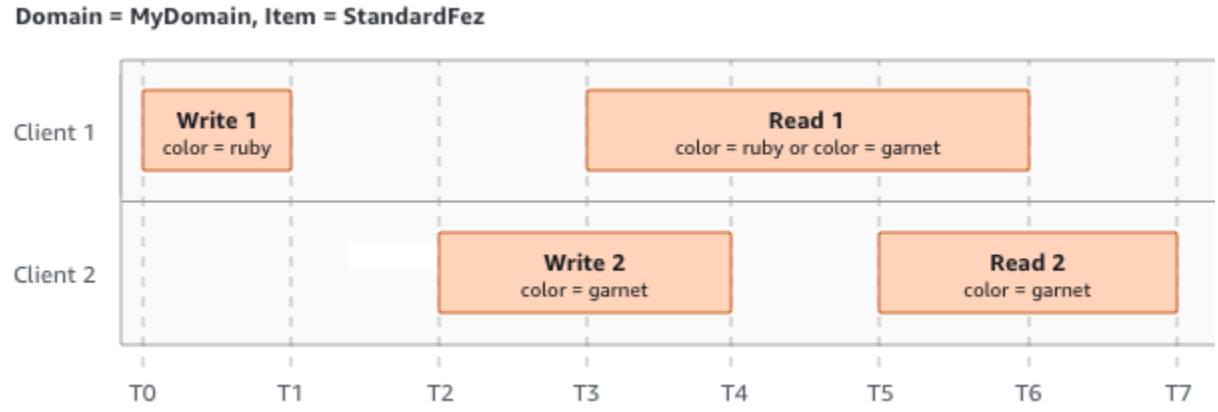
Aplicaciones simultáneas

En esta sección se proporcionan ejemplos del comportamiento que se espera de Amazon S3 cuando varios clientes escriben en los mismos elementos.

En este ejemplo, tanto W1 (escritura 1) como W2 (escritura 2) se completan antes del inicio de R1 (lectura 1) y R2 (lectura 2). Debido a que S3 es altamente consistente, tanto R1 como R2 devuelven `color = ruby`.



En el siguiente ejemplo, W2 no se completa antes del inicio de R1. Por lo tanto, R1 podría devolver `color = ruby` o `color = garnet`. Sin embargo, dado que W1 y W2 finalizan antes del inicio de R2, R2 devuelve `color = garnet`.



En el último ejemplo, W2 se inicia antes de que W1 haya recibido una confirmación. Por lo tanto, estas escrituras se consideran simultáneas. Amazon S3 utiliza internamente la semántica “last-writer-wins” (el último en escribir gana) para determinar qué escritura tiene prioridad. Sin embargo, el orden en el que Amazon S3 recibe las solicitudes y el orden en el que las aplicaciones envían las confirmaciones no se puede predecir debido a factores como la latencia de la red. Por ejemplo, W2 puede ser iniciado por una instancia de Amazon EC2 en la misma región, mientras que W1 podría ser iniciado por un host que está más lejos. La mejor manera de determinar el valor final es realizar una lectura después de que se ha recibido la confirmación de ambas escrituras.

Servicios relacionados

Una vez que carga sus datos en Amazon S3, puede utilizarlos con otros servicios de AWS. Los siguientes servicios son los que puede utilizar con más frecuencia:

- Amazon Elastic Compute Cloud (Amazon EC2): proporciona capacidad de computación escalable y segura en Nube de AWS. El uso de Amazon EC2 elimina la necesidad de invertir inicialmente en hardware, de manera que puede desarrollar e implementar aplicaciones en menos tiempo. Puede usar Amazon EC2 para lanzar tantos servidores virtuales como necesite, configurar la seguridad y las redes, y administrar el almacenamiento.
- Amazon EMR: ayuda a las empresas, investigadores, analistas de datos y desarrolladores procesar de forma fácil y rentable grandes volúmenes de datos. Amazon EMR utiliza un marco Hadoop alojado que se ejecuta en la infraestructura basada en la web de Amazon EC2 y Amazon S3.
- AWSFamilia Snow: ayuda a los clientes que necesitan ejecutar operaciones en entornos austeros, no pertenecientes al centro de datos, y en ubicaciones en las que no existe una conectividad de red coherente. Puede usar dispositivos de Snow Family AWS para acceder localmente y de manera rentable al almacenamiento y la potencia informática de Nube de AWS en lugares en los que posiblemente no haya una conexión a Internet.
- AWS Transfer Family: proporciona compatibilidad totalmente administrada para transferencias de archivos directamente desde Amazon S3 o Amazon Elastic File System (Amazon EFS) mediante el protocolo de transferencia de archivos (SFTP) de Secure Shell (SSH), el protocolo de transferencia de archivos a través de SSL (FTPS) y el protocolo de transferencia de archivos (FTP).

Acceso a Amazon S3

Puede trabajar con Amazon S3 de cualquiera de las siguientes formas:

AWS Management Console

La consola es una interfaz de usuario basada en la web para administrar Amazon S3 y los recursos de AWS. Si se ha registrado para Cuenta de AWS, puede acceder a la consola de Amazon S3 iniciando sesión en AWS Management Console y eligiendo S3 en la página de inicio de AWS Management Console.

AWS Command Line Interface

Puede utilizar elAWSHerramientas de línea de comandos para emitir comandos o compilar scripts en la línea de comandos de su sistema con el fin de ejecutarAWS(incluidas las tareas S3).

AWS Command Line Interface (AWS CLI) proporciona comandos para una amplia gama de Servicios de AWS. La AWS CLI es compatible con Windows, macOS y Linux. Para empezar, consulte la AWS Command Line Interface Guía de usuario de . Para obtener más información acerca de los comandos de Amazon S3, consulte s3api y s3control en la AWS CLIReferencia de los comandos de la .

SDK de AWS

AWS ofrece SDK (kits de desarrollo de software) que se componen de bibliotecas y código de muestra para diversos lenguajes de programación y plataformas (Java, Python, Ruby, .NET, iOS, Android, etc.). Los AWS SDK proporcionan una forma cómoda de crear acceso a S3 y AWS. Amazon S3 es un servicio de REST. Puede enviar solicitudes a Amazon S3 usando las bibliotecas de SDK de AWS, que envuelven la API de REST de Amazon S3 subyacentes y simplifican sus tareas de programación. Por ejemplo, los SDK se encargan de tareas como calcular firmas, firmar solicitudes criptográficamente, gestionar los errores y reintentar las solicitudes de forma automática. Para obtener información sobre los SDK de AWS (por ejemplo: cómo descargarlos e instalarlos), consulte Herramientas para AWS.

Toda interacción con Amazon S3 es o autenticada o anónima. Si utiliza el SDK de AWS, las bibliotecas computan la firma para la autenticación a partir de las claves que usted proporciona. Para obtener más información acerca de cómo realizar solicitudes a Amazon S3, consulteRealizar solicitudes.

API de REST de Amazon S3

La arquitectura de Amazon S3 está diseñada con un lenguaje de programación neutro y utiliza interfaces admitidas por AWS para almacenar y recuperar objetos. Puede acceder a S3 yAWSmediante programación mediante la API de REST de Amazon S3. La API de REST es una interfaz HTTP para Amazon S3. Con la API de REST, usted puede utilizar solicitudes HTTP estándar para crear, recuperar y eliminar buckets y objetos.

Puede utilizar cualquier conjunto de herramientas que admita HTTP para utilizar la API de REST. Incluso puede utilizar un navegador para recuperar objetos, siempre y cuando se puedan leer de forma anónima.

La API de REST utiliza códigos de estado y encabezados HTTP estándar, para que los conjuntos de herramientas y los navegadores estándar funcionen según lo previsto. En algunas áreas, hemos añadido una funcionalidad al HTTP (por ejemplo: añadimos encabezados para admitir el control de acceso). En estos casos, hicimos todo lo posible para añadir la nueva funcionalidad de manera que coincida con el estilo del uso de HTTP estándar.

Si realiza llamadas directas a la API de REST en su aplicación, debe escribir el código para computar la firma y añadirla a la solicitud. Para obtener más información acerca de cómo realizar solicitudes a Amazon S3, consulteRealizar solicitudes.

> **Note**
>
> La compatibilidad con la API de SOAP por HTTP está obsoleta, pero aún se encuentra disponible con HTTPS. Las características más recientes de Amazon S3 no son compatibles con SOAP. Le recomendamos que utilice la API de REST o los SDK de AWS.

Pago de Amazon S3

Los precios de Amazon S3 están diseñados de manera que no tenga que planificar los requisitos de almacenamiento de su aplicación. La mayoría de los proveedores de almacenamiento requieren que adquiera una cantidad predeterminada de almacenamiento y capacidad de transferencia de red. En este escenario, si excede esa capacidad, su servicio se cancela o usted debe pagar cargos excesivos. Si no excede esa capacidad, paga como si la hubiera utilizado toda.

Amazon S3 le cobra solo lo que realmente utiliza, sin costes ocultos ni cargos excesivos. Esto ofrece a los desarrolladores un servicio de costo variable que puede crecer junto con sus empresas

mientras disfrutan de las ventajas de costos que ofrece la infraestructura de AWS. Para obtener más información, consulte Precios de Amazon S3.

Cuando se registra en AWS, su Cuenta de AWS se registra automáticamente en todos los servicios de AWS, incluido Amazon S3. No obstante, solo se le cobrará por los servicios que utilice. Si es cliente nuevo de Amazon S3, puede comenzar con Amazon S3 de forma gratuita. Para obtener más información, consulte AWS capa gratuita.

Para ver su factura, vaya al Panel de Billing and Cost Management en la consola de AWS Billing and Cost Management. Para obtener más información sobre cómo usar Cuenta de AWS, consulte la Guía del usuario de AWS Billing. Si tiene alguna pregunta sobre los eventos, las cuentas y la facturación de AWS, Cuentas de AWSpóngase en contacto con AWS Support.

Conformidad con DSS PCI

Amazon S3 admite el procesamiento, el almacenamiento y la transmisión de datos de tarjetas de crédito por parte de un comerciante o proveedor de servicios y se ha validado por estar conforme con el Estándar de Seguridad de Datos para la Industria de Tarjeta de Pago (PCI DSS). Para obtener más información acerca de PCI DSS, incluido cómo solicitar una copia del Paquete de conformidad con PCI de AWS, consulte PCI DSS Nivel 1.

Introducción a Amazon S3

Puede comenzar a utilizar Amazon S3 al trabajar con buckets y objetos. Un bucket es un contenedor de objetos. Un objeto es un archivo y cualquier metadato que describa ese archivo.

Para almacenar un objeto en Amazon S3, cree un bucket y, a continuación, cargue el objeto en el bucket. Cuando el objeto está en el bucket, puede abrirlo, descargarlo y moverlo. Cuando ya no necesite un objeto o un bucket, puede limpiar los recursos.

Con Amazon S3 paga únicamente por lo que usa. Para obtener más información acerca de las características y precios de Amazon S3, consulte Amazon S3. Si es cliente nuevo de Amazon S3, puede comenzar con Amazon S3 de forma gratuita. Para obtener más información, consulte Capa gratuita de AWS.

> **Note**
>
> Para obtener más información sobre el uso de la clase de almacenamiento Amazon S3 Express One Zone con buckets de directorio, consulte ¿Qué es S3 Express One Zone? y Buckets de directorio.

Vídeo: Introducción a Amazon S3

Requisitos previos

Antes de comenzar, asegúrese de que ha realizado los pasos que se detallan en Requisito previo: Configuración de Amazon S3.

Temas

Requisito previo: Configuración de Amazon S3

Cuando se registra en AWS, su Cuenta de AWS se registra automáticamente en todos los servicios de AWS, incluido Amazon S3. Solo se le cobrará por los servicios que utilice.

Con Amazon S3 paga únicamente por lo que usa. Para obtener más información acerca de las características y precios de Amazon S3, consulte Amazon S3. Si es cliente nuevo de Amazon S3, puede comenzar con Amazon S3 de forma gratuita. Para obtener más información, consulte Capa gratuita de AWS.

Para configurar Amazon S3, siga los pasos descritos en las secciones siguientes.

Cuando se registra en AWS y configura Amazon S3, puede cambiar de forma opcional el idioma de visualización de la AWS Management Console. Para obtener más información, consulte Cambio del idioma de la AWS Management Console en la Guía de introducción de la AWS Management Console.

Temas

- Registro en una Cuenta de AWS
- Cree un usuario con acceso de administrador

Registro en una Cuenta de AWS

Si no dispone de una Cuenta de AWS, siga estos pasos para crear una.

Procedimiento para registrarse en Cuenta de AWS

1. Abra https://portal.aws.amazon.com/billing/signup.
2. Siga las instrucciones que se le indiquen.

   Parte del procedimiento de registro consiste en recibir una llamada telefónica e indicar un código de verificación en el teclado del teléfono.

   Al registrarse en una Cuenta de AWS, se crea un Usuario raíz de la cuenta de AWS. El usuario raíz tendrá acceso a todos los Servicios de AWS y recursos de esa cuenta. Como práctica recomendada de seguridad, asigne acceso de administrador a un usuario y utilice únicamente el usuario raíz para ejecutar tareas que requieren acceso de usuario raíz.

AWS le enviará un email de confirmación cuando complete el proceso de registro. Puede ver la actividad de la cuenta y administrar la cuenta en cualquier momento entrando en https://aws.amazon.com/ y seleccionando Mi cuenta.

Cree un usuario con acceso de administrador

Después de registrarse para obtener una Cuenta de AWS, proteja su Usuario raíz de la cuenta de AWS, habilite AWS IAM Identity Center y cree un usuario administrativo para no utilizar el usuario raíz en las tareas cotidianas.

Protección de Usuario raíz de la cuenta de AWS

1. Inicie sesión en AWS Management Console como propietario de la cuenta; para ello, elija Usuario raíz e introduzca el correo electrónico de su Cuenta de AWS. En la siguiente página, escriba su contraseña.

   Para obtener ayuda para iniciar sesión con el usuario raíz, consulte Signing in as the root user en la Guía del usuario de AWS Sign-In.
2. Active la autenticación multifactor (MFA) para el usuario raíz.

   Para obtener instrucciones, consulte Habilitación de un dispositivo MFA virtual para su usuario raíz de la Cuenta de AWS (consola) en la Guía del usuario de IAM.

Cree un usuario con acceso de administrador

1. Activar IAM Identity Center.

   Consulte las instrucciones en Activar AWS IAM Identity Center en la Guía del usuario de AWS IAM Identity Center.
2. En IAM Identity Center, conceda acceso administrativo a un usuario.

   Para ver un tutorial sobre cómo utilizar Directorio de IAM Identity Center como origen de identidad, consulte Configuración del acceso de los usuarios con el Directorio de IAM Identity Center predeterminado en la Guía del usuario de AWS IAM Identity Center.

Iniciar sesión como usuario con acceso de administrador

- Para iniciar sesión con el usuario de IAM Identity Center, utilice la URL de inicio de sesión que se envió a la dirección de correo electrónico cuando creó el usuario de IAM Identity Center.

Para obtener ayuda para iniciar sesión con un usuario del IAM Identity Center, consulte Inicio de sesión en el portal de acceso de AWS en la Guía del usuario de AWS Sign-In.

Asignar acceso a usuarios adicionales

1. En IAM Identity Center, cree un conjunto de permisos que siga la práctica recomendada de aplicar permisos de privilegios mínimos.

   Para conocer las instrucciones, consulte Creación de un conjunto de permisos en la Guía del usuario de AWS IAM Identity Center.

2. Asigne usuarios a un grupo y, a continuación, asigne el acceso de inicio de sesión único al grupo.

   Para conocer las instrucciones, consulte Agregar grupos en la Guía del usuario de AWS IAM Identity Center.

Paso 1: Crear su primer bucket de S3

Después de registrarse en AWS, estará listo para crear un bucket en Amazon S3 a través de la AWS Management Console. Todos los objetos de Amazon S3 se almacenan en un bucket. Por tanto, debe crear un bucket para poder almacenar datos en Amazon S3.

> **Note**
>
> Para obtener más información sobre el uso de la clase de almacenamiento Amazon S3 Express One Zone con buckets de directorio, consulte ¿Qué es S3 Express One Zone? y Buckets de directorio.

> **Note**
>
> No se le cobrará por la creación de un bucket. Solo se le cobrará por almacenar objetos en el bucket y por transferirlos dentro y fuera de este. Los cargos en los que incurrirá al seguir los ejemplos de esta guía son mínimos (inferiores a 1 USD). Para obtener más información acerca de los cargos de almacenamiento, consulte Precios de Amazon S3.

1. Inicie sesión AWS Management Console Management Console y abra la consola de Amazon S3 en https://console.aws.amazon.com/s3/.
2. En la barra de navegación de la parte superior de la página, elija el nombre de la Región de AWS que aparece. A continuación, elija en Region (Región) la región en la que desea crear un bucket.

   > **Note**
   >
   > Puede seleccionar una región cercana para minimizar la latencia y los costos, así como para satisfacer los requisitos normativos. Los objetos almacenados en una región nunca abandonarán esa región salvo que usted los transfiera de forma específica a otra. Para una lista de Regiones de AWS de Amazon S3, consulte Puntos de conexión de Servicio de AWS en la Referencia general de Amazon Web Services.

3. En el panel de navegación izquierdo, elija Instancias.
4. Elija Crear bucket.

   Se abrirá la página Crear bucket.
5. En General configuration (Configuración general), vea la Región de AWS donde se creará el bucket.
6. En Bucket type (Tipo de depósito), seleccione General purpose (Uso general).
7. En Nombre del bucket, escriba un nombre para el bucket.

   El nombre del bucket debe:

   - Ser exclusivo dentro de una partición. Una partición es una agrupación de regiones. AWS actualmente tiene tres particiones: `aws` (regiones estándar), `aws-cn` (regiones de China) y `aws-us-gov` (AWS GovCloud (US) Regions).
   - Tener entre 3 y 63 caracteres.
   - Consistir únicamente de letras minúsculas, números, puntos (.) y guiones (-). Para obtener una mejor compatibilidad, se recomienda evitar el uso de puntos (.) en los nombres de los buckets, excepto para los buckets que se utilizan únicamente para el alojamiento estático de sitios web.
   - Comenzar y terminar por un número o una letra.

Una vez que haya creado el bucket, no podrá modificar su nombre. Para obtener más información sobre la nomenclatura de los buckets, consulte Reglas de nomenclatura de buckets.

> **Important**
>
> Evite incluir información confidencial, como números de cuenta, en el nombre del bucket. El nombre del bucket será visible en las URL que señalan a los objetos almacenados en él.

8. AWS Management Console le permite copiar la configuración de un bucket existente en el nuevo bucket. Si no desea copiar la configuración de un bucket existente, vaya al paso siguiente.

> **Note**
>
> Esta opción:
>
> - No está disponible en la AWS CLI y solo está disponible en la consola
> - No está disponible para buckets de directorio
> - No copia la política de bucket del bucket existente al nuevo bucket

Para copiar la configuración de un bucket existente, en Copy settings from existing Bucket (Copiar la configuración del depósito existente), seleccione Choose Bucket (Elegir bucket). Se abre la ventana Choose bucket (Elegir bucket). Busque el bucket con los ajustes que quiera copiar y seleccione Choose bucket (Elegir bucket). Se cierra la ventana Choose bucket (Elegir bucket) y se vuelve a abrir la ventana Create bucket (Crear bucket).

En Copy settings from existing bucket (Copiar la configuración del bucket existente), ahora verá el nombre del bucket que ha seleccionado. También verá la opción Restore defaults (Restaurar los valores predeterminados) que puede usar para eliminar la configuración del bucket copiada. Revise la configuración restante del bucket en la página Create bucket (Crear bucket). Verá que ahora coinciden con la configuración del bucket que seleccionó. Puede saltar al paso final.

9. En Propiedad de objetos, para desactivar o habilitar las ACL y controlar la propiedad de los objetos cargados en el bucket, elija una de las siguientes configuraciones:

ACL desactivadas

- Propietario del bucket obligatorio (predeterminado): las ACL están desactivadas y el propietario del bucket tiene automáticamente la propiedad y el control total sobre cada objeto del bucket. Las ACL ya no afectan a los permisos de acceso de los datos del bucket de S3. El bucket utiliza políticas exclusivamente para definir el control de acceso.

  Las ACL están desactivadas de forma predeterminada. La mayoría de los casos de uso modernos de Amazon S3 ya no requieren el uso de ACL. Le recomendamos que mantenga las ACL desactivadas, excepto en circunstancias inusuales en las que necesite controlar el acceso a cada objeto de manera individual. Para obtener más información, consulte Control de la propiedad de los objetos y desactivación de las ACL del bucket.

ACL habilitadas

- Propietario del bucket preferido: el propietario del bucket tiene la propiedad y el control total sobre los nuevos objetos que otras cuentas escriben en el bucket con la ACL predefinida `bucket-owner-full-control`.

  Si aplica la configuración de propietario del bucket preferido para requerir que todas las cargas de Amazon S3 incluyan la ACL predefinida `bucket-owner-full-control`, puede agregar una política de bucket que solo permita cargas de objetos que utilicen esta ACL.
- Escritor del objeto: la Cuenta de AWS que carga un objeto es propietaria del objeto, tiene control total sobre él y puede conceder a otros usuarios acceso a él a través de ACL.

> **Note**
>
> La configuración predeterminada es Aplicada al propietario del bucket. Para aplicar la configuración predeterminada y mantener las ACL deshabilitadas, solo se necesita el permiso `s3:CreateBucket`. Para habilitar las ACL, debe tener el permiso `s3:PutBucketOwnershipControls`.

10. En Configuración de bloqueo de acceso público para este bucket, elija la configuración Bloquear acceso público que desee aplicar al bucket.

    De forma predeterminada, las cuatro configuraciones de Bloqueo de acceso público estarán activas. Le recomendamos que deje todas las configuraciones activadas a menos que sepa

que necesita desactivar una o varias para su caso de uso específico. Para obtener más información acerca del bloqueo del acceso público, consulte Bloquear el acceso público a su almacenamiento de Amazon S3.

> **Note**
>
> Para habilitar todas las configuraciones de Bloqueo de acceso público, solo se requiere el permiso `s3:CreateBucket`. Para desactivar cualquier configuración de Bloqueo de acceso público, debe tener el permiso `s3:PutBucketPublicAccessBlock`.

11. (Opcional) En Control de versiones de buckets, puede elegir si desea mantener variantes de objetos en su bucket. Para obtener más información sobre el control de versiones, consulte Usar el control de versiones en buckets de S3.

    Para deshabilitar o habilitar el control de versiones en su bucket, elija Disable (Deshabilitar) o Enable (Habilitar).

12. (Opcional) En Tags (Etiquetas), puede elegir añadir etiquetas a su bucket. Las etiquetas son pares clave-valor que se utilizan para categorizar el almacenamiento de información.

    Para agregar una etiqueta de bucket, introduzca un valor en Clave y opcionalmente otro en Valor y elija Añadir etiqueta.

13. En Cifrado predeterminado, elija Editar.

14. Para configurar el cifrado predeterminado, en Tipo de cifrado, elija una de las siguientes opciones:

    - Clave administrada de Amazon S3 (SSE-S3)
    - Clave de AWS Key Management Service (SSE-KMS)

> **Important**
>
> Si utiliza la opción de SSE-KMS para la configuración de cifrado predeterminado, se le aplicará la cuota de solicitudes por segundo (RPS) de AWS KMS. Para obtener más información acerca de las cuotas de AWS KMS y cómo solicitar un aumento de cuota, consulte Cuotas en la Guía para desarrolladores de AWS Key Management Service.

Los buckets y los objetos nuevos se cifran mediante el cifrado del lado del servidor con una clave administrada de Amazon S3 como nivel básico de configuración de cifrado. Para obtener más información acerca del cifrado predeterminado, consulte Establecer el comportamiento del cifrado predeterminado del lado del servidor para los buckets de Amazon S3.

Para obtener más información sobre el uso del cifrado del lado del servidor de Amazon S3 para cifrar los datos, consulte Uso del cifrado del servidor con claves administradas por Amazon S3 (SSE-S3).

15. Si ha elegido la clave de AWS Key Management Service (SSE-KMS), haga lo siguiente:

    a. En Clave de AWS KMS, especifique su clave de KMS de una de las siguientes maneras:

    - Para seleccionar de una lista de claves de KMS disponibles, marque Elija entre sus claves de AWS KMS keys y seleccione su clave de KMS de la lista de claves disponibles.

      En esta lista aparecen tanto la Clave administrada de AWS (`aws/s3`) como las claves administradas por el cliente. Para obtener más información acerca de las claves administradas por el cliente, consulte Claves de cliente y claves de AWS en la Guía para desarrolladores de AWS Key Management Service.
    - Para introducir el ARN de la clave de KMS, elija Introducir el ARN de la AWS KMS key e introduzca el ARN de la clave de KMS en el campo que aparece.
    - Para crear una nueva clave administrada por el cliente en la consola de AWS KMS, elija Crear una clave de KMS.

      Para obtener más información acerca de cómo crear una AWS KMS key, consulte Creación de claves en la AWS Key Management Service Guía para desarrolladores.

> **Important**
>
> Solo puede utilizar las claves de KMS que estén disponibles en la misma Región de AWS del bucket. La consola de Amazon S3 solo muestra las primeras 100 claves de KMS de la misma región del bucket. Para utilizar una clave de KMS que no aparezca en la lista, debe introducir el ARN de la clave de KMS. Si desea utilizar una clave de KMS propiedad de una cuenta de diferente, primero debe tener permiso para utilizar la clave y, después, debe introducir el ARN de la clave de KMS. Para obtener más información sobre los permisos entre cuentas para las

claves de KMS, consulte Crear claves de KMS que otras cuentas puedan utilizar en la Guía para desarrolladores de AWS Key Management Service. Para obtener más información sobre SSE-KMS, consulte Especificación del cifrado del lado del servidor con AWS KMS (SSE-KMS).

Cuando utilice una AWS KMS key para el cifrado en el lado del servidor en Amazon S3, debe elegir una clave de cifrado de KMS simétrica. Amazon S3 admite solo claves de KMS de cifrado simétricas y no claves de KMS asimétricas. Para obtener más información, consulte Identificación de claves de KMS simétricas y asimétricas en la Guía para desarrolladores de AWS Key Management Service.

Para obtener más información acerca de cómo crear una AWS KMS key, consulte Creación de claves en la AWS Key Management Service Guía para desarrolladores. Para obtener más información acerca del uso de AWS KMS con Amazon S3, consulte Uso del cifrado del servidor con claves de AWS KMS (SSE-KMS).

b. Cuando configure el bucket para que use el cifrado predeterminado con SSE-KMS, también puede habilitar las claves de bucket de S3. Las claves de bucket de S3 reducen el costo del cifrado al reducir el tráfico de solicitudes de Amazon S3 a AWS KMS. Para obtener más información, consulte Reducción del costo de SSE-KMS con las claves de bucket de Amazon S3.

   Para utilizar las claves de bucket de S3, en Clave de bucket, seleccione Habilitar.

16. (Opcional) Si desea habilitar el bloqueo de objetos en S3, haga lo siguiente:

    a. Seleccione Advanced settings (Ajustes avanzados).

> **Important**
>
> Al habilitar Bloqueo de objetos, también se habilita el control de versiones para el bucket. Después de habilitar, debe configurar la retención predeterminada de Object Lock y la configuración de retención legal para evitar que los nuevos objetos se eliminen o se sobrescriban.

    b. Si desea habilitar el bloqueo de objetos, elija Enable (Habilitar), lea la advertencia que aparece y acéptela.

Para obtener más información, consulte Usar Bloqueo de objetos de S3.

> **Note**
>
> Para crear un bucket con Bloqueo de objetos, debe tener los siguientes permisos: `s3:CreateBucket`, `s3:PutBucketVersioning` y `s3:PutBucketObjectLockConfiguration`.

17. Elija Crear bucket.

Ha creado un bucket en Amazon S3.

Paso siguiente

Para agregar un objeto al bucket, consulte Paso 2: Cargar un objeto en el bucket.

Paso 2: Cargar un objeto en el bucket

Después de crear un bucket en Amazon S3, podrá cargar un objeto en el bucket. Un objeto puede ser cualquier clase de archivo: un archivo de texto, una fotografía, un video, etc.

> **Note**
>
> Para obtener más información sobre el uso de la clase de almacenamiento Amazon S3 Express One Zone con buckets de directorio, consulte ¿Qué es S3 Express One Zone? y Buckets de directorio.

Para cargar un objeto a un bucket

1. Abra la consola de Amazon S3 en https://console.aws.amazon.com/s3.
2. En la lista Buckets, seleccione el nombre del bucket en el que desee cargar el objeto.
3. En la pestaña Objects (Objetos) del bucket, elija Upload (Cargar).
4. En Files and folders (Archivos y carpetas), elija Add files (Añadir archivos).
5. Seleccione un archivo que cargar y luego seleccione Open (Abrir).
6. Seleccione Cargar.

Ha añadido correctamente un objeto a su bucket.

Paso siguiente

Para ver el objeto, consulte Paso 3: Descargar un objeto.

Paso 3: Descargar un objeto

Ahora que ha cargado un objeto a un bucket, puede ver información sobre el objeto y descargarlo en su equipo local.

> **Note**
>
> Para obtener más información sobre el uso de la clase de almacenamiento Amazon S3 Express One Zone con buckets de directorio, consulte ¿Qué es S3 Express One Zone? y Buckets de directorio.

Uso de la consola de S3

En esta sección se explica cómo utilizar la consola de Amazon S3 para descargar un objeto de un bucket de S3.

> **Note**
>
> - Solo se puede descargar un objeto a la vez.
> - Si utiliza la consola de Amazon S3 para descargar un objeto cuyo nombre de clave termine con un punto (`.`), se eliminará el punto del nombre de clave del objeto descargado. Para conservar el punto al final del nombre del objeto descargado, debe usar la AWS Command Line Interface (AWS CLI), los SDK de AWS o la API de REST de Amazon S3.

Para descargar un objeto desde un bucket de S3

1. Inicie sesión en la AWS Management Console y abra la consola de Amazon S3 en https://console.aws.amazon.com/s3/.
2. En la lista Buckets (Buckets), elija el nombre del bucket que contiene el objeto que desea descargar.
3. Puede descargar un objeto de un bucket de S3 de cualquiera de las siguientes maneras:

- Seleccione la casilla de verificación situada junto al objeto y elija Descargar. Si desea descargar el objeto a una carpeta específica, en el menú Acciones, seleccione Descargar como.
- Si desea descargar una versión específica del objeto, active Mostrar versiones (situado junto al cuadro de búsqueda). Seleccione la casilla de verificación situada junto a la versión del objeto que desee y elija Descargar. Si desea descargar el objeto a una carpeta específica, en el menú Acciones, seleccione Descargar como.

Ha descargado correctamente el objeto.

Paso siguiente

Para copiar y pegar el objeto en Amazon S3, consulte Paso 4: Copiar el objeto en una carpeta.

Paso 4: Copiar el objeto en una carpeta

Ha añadido un objeto a un bucket y ha descargado el objeto. Ahora, cree una carpeta y copie el objeto y péguelo en la carpeta.

> **Note**
>
> Para obtener más información sobre el uso de la clase de almacenamiento Amazon S3 Express One Zone con buckets de directorio, consulte ¿Qué es S3 Express One Zone? y Buckets de directorio.

Para copiar un objeto en una carpeta

1. En la lista de Buckets, elija el nombre del bucket.
2. Elija Create folder (Crear carpeta) y configure una carpeta nueva:
   a. Escriba un nombre para la carpeta (por ejemplo, `favorite-pics`).
   b. Para la configuración de cifrado de carpeta, elija Disable (Desactivar).
   c. Seleccione Guardar.
3. Desplácese hasta el bucket o la carpeta de Amazon S3 que contiene los objetos que desea copiar.

4. Seleccione la casilla de verificación situada a la izquierda de los nombres de los objetos que desea copiar.
5. Elija Actions (Acciones) y luego Copy (Copiar) en la lista de opciones que aparece.

   También puede elegir Copy (Copiar) en las opciones de la esquina superior derecha.
6. Elija la carpeta de destino:

   a. Elija Browse S3 (Examinar S3).
   b. Elija el botón de opción a la izquierda del nombre de la carpeta.

      Para navegar por una carpeta y elegir una subcarpeta como destino, elija el nombre de la carpeta.
   c. Elija Choose destination (Elegir destino).

   La ruta de acceso a la carpeta de destino aparece en el cuadro Destination (Destino). En Destination (Destino), puede escribir también la ruta de destino, por ejemplo, s3://*nombre-bucket*/*nombre-carpeta*/.
7. En la parte inferior derecha, elija Copy (Copiar).

   Amazon S3 copia los objetos en la carpeta de destino.

Paso siguiente

Para eliminar un objeto y un bucket en Amazon S3, consulte Paso 5: Eliminar los objetos y el bucket.

Paso 5: Eliminar los objetos y el bucket

Cuando ya no necesite un objeto o un bucket, le recomendamos que los elimine para evitar que se carguen más. Si ha completado esta explicación introductoria como un ejercicio de aprendizaje y no piensa utilizar el bucket o los objetos, le recomendamos los elimine para que no se acumulen los cargos.

Antes de eliminar el bucket, debe vaciarlo o eliminar los objetos que contiene. Después de eliminar los objetos y el bucket, ya no estarán disponibles.

Si desea seguir utilizando el mismo nombre de bucket, le recomendamos que elimine los objetos o vacíe el bucket, pero no lo elimine. Después de eliminar un bucket, el nombre estará disponible para

reutilizarlo. Sin embargo, otra Cuenta de AWS podría crear un bucket con el mismo nombre antes de tener la oportunidad de reutilizarlo.

> **Note**
>
> Para obtener más información sobre el uso de la clase de almacenamiento Amazon S3 Express One Zone con buckets de directorio, consulte ¿Qué es S3 Express One Zone? y Buckets de directorio.

Temas

Eliminación de un objeto

Si desea elegir los objetos que desea eliminar sin vaciar todos los objetos del bucket, puede eliminar un objeto.

1. En la lista Buckets, seleccione el nombre del bucket que contiene el objeto que desea eliminar.
2. Seleccione los objetos que desea eliminar.
3. Elija Eliminar en las opciones de arriba a la derecha.
4. En la página Eliminar objetos, escriba **`delete`** para confirmar la eliminación de los objetos.
5. Elija Eliminar objetos.

Vaciar el bucket

Si planea eliminar el bucket, primero debe vaciarlo, y se eliminarán todos los objetos del bucket.

Para vaciar un bucket

1. En la lista Buckets, elija el bucket que desee vaciar y, a continuación, elija Vaciar.
2. Para confirmar que desea vaciar el bucket y eliminar todos los objetos que contiene, en Empty bucket (Vaciar bucket), escriba **`permanently delete`** (borrar de forma definitiva).

> **Important**
>
> La operación de vaciado del bucket no se puede revertir. Se eliminarán los objetos añadidos al bucket mientras la acción de vaciado del bucket está en curso.

3. Para vaciar el bucket y eliminar todos los objetos que contiene, elija Vaciar.

   Se abrirá una página Vaciar bucket: Estado que puede utilizar para revisar un resumen de las eliminaciones de objetos que han fallado y que se han realizado correctamente.

4. Para volver a la lista de buckets, seleccione Salir.

Eliminar el bucket

Después de vaciar el bucket o eliminar todos los objetos de él, puede eliminarlo.

1. Para eliminar un bucket, selecciónelo en la lista Buckets.
2. Elija Eliminar.
3. Para confirmar la eliminación, en Delete bucket (Eliminar bucket), escriba el nombre del bucket.

> **Important**
>
> La eliminación de un bucket no se puede revertir. Los nombres de bucket son únicos. Si elimina el bucket, otro usuario de AWS podrá utilizar el nombre. Si desea seguir utilizando el mismo nombre de bucket, no elimine el bucket. En su lugar, vacíe y guarde el bucket.

4. Para eliminar el bucket, elija Eliminar bucket.

Siguientes pasos

En los ejemplos anteriores, usted aprendió a realizar algunas tareas básicas de Amazon S3.

En los siguientes temas se explican las rutas de aprendizaje que puede usar para obtener más información acerca de Amazon S3 para poder implementarlo en sus aplicaciones.

> **Note**
>
> Para obtener más información sobre el uso de la clase de almacenamiento Amazon S3 Express One Zone con buckets de directorio, consulte ¿Qué es S3 Express One Zone? y Buckets de directorio.

Temas

Comprender los casos de uso frecuentes

Puede utilizar Amazon S3 para admitir su caso de uso específico. La AWSBiblioteca de soluciones y el AWSBlog proporcionan información específica sobre casos de uso y tutoriales. A continuación, se indican algunos casos de uso frecuentes para Amazon S3:

- Copia de seguridad y almacenamiento: utilice las características de administración de almacenamiento de Amazon S3 para administrar costos, cumplir con los requisitos normativos, reducir la latencia y guardar varias copias distintas de los datos para cumplir los requisitos de conformidad.
- Alojamiento de aplicaciones: implemente, instale y administre aplicaciones web confiables, altamente escalables y de bajo costo. Por ejemplo, puede configurar el bucket de Amazon S3 para alojar un sitio web estático. Para obtener más información, consulte Alojamiento de un sitio web estático mediante Amazon S3.
- Alojamiento multimedia: cree una infraestructura de alta disponibilidad que aloje cargas y descargas de videos, fotos o música.
- Entrega de software: aloje sus aplicaciones de software para que los clientes puedan descargarlas.

Controle el acceso a sus buckets y objetos

Amazon S3 proporciona una variedad de características y herramientas de seguridad. Para obtener una descripción general, consulte Administración de accesos.

De forma predeterminada, los buckets y los objetos de S3 son privados. Solo tiene acceso a los recursos de S3 que cree. Puede utilizar las siguientes características para conceder permisos de recursos pormenorizados que admitan su caso de uso específico o para auditar los permisos de sus recursos de Amazon S3.

- S3 Block Public Access: bloquee el acceso público a los buckets y objetos de S3. De forma predeterminada, la configuración de bloqueo del acceso público se activa en el nivel de bucket.
- Identidades de AWS Identity and Access Management (IAM): utilice IAM o AWS IAM Identity Center para crear identidades de IAM en la Cuenta de AWS para administrar el acceso a los recursos de Amazon S3. Por ejemplo, puede usar IAM con Amazon S3 para controlar el tipo de acceso que tiene un usuario o un grupo de usuarios a un bucket de Amazon S3 que es propiedad de su Cuenta de AWS. Para obtener más información acerca de las identidades y prácticas recomendadas de IAM, consulte Identidades (usuarios, grupos de usuarios y roles) de IAM en la Guía del usuario de IAM.
- Políticas de buckets: utilice el lenguaje de políticas basado en IAM para configurar permisos basados en recursos para los buckets de S3 y los objetos en ellos.
- Listas de control de acceso (ACL)— Conceder permisos de lectura y escritura para buckets y objetos individuales a usuarios autorizados. Como regla general, se recomienda utilizar políticas basadas en recursos de S3 (políticas de bucket y políticas de punto de acceso) o políticas de usuario de IAM para el control de acceso en lugar de las ACL. Las políticas son una opción de control de acceso simplificada y más flexible. Con las políticas de bucket y las políticas de puntos de acceso, puede definir reglas que se apliquen ampliamente a todas las solicitudes a sus recursos de Amazon S3. Para obtener más información acerca de casos específicos en que usaría ACL en lugar de políticas basadas en recursos o políticas de usuarios de IAM, consulte Administración de identidades y accesos para Amazon S3.
- S3 Object Ownership: tome posesión de cada objeto del bucket, lo que simplificará la administración del acceso a los datos almacenados en Amazon S3. S3 Object Ownership es una configuración en el nivel de bucket de Amazon S3 que puede usar para desactivar o activar las ACL. Las ACL están desactivadas de forma predeterminada. Cuando las ACL están desactivadas, el propietario del bucket posee todos los objetos del bucket y administra el acceso a los datos de forma exclusiva mediante políticas de administración de acceso.

- Analizador de acceso de IAM para S3: evalúe y monitoree sus políticas de acceso al bucket de S3, asegurándose de que las políticas solo proporcionen el acceso previsto a sus recursos de S3.

Administre y monitoree su almacenamiento

- Administrar su almacenamiento: después de crear buckets y cargar objetos en Amazon S3, puede administrar el almacenamiento de objetos. Por ejemplo, puede utilizar el control de versiones de S3 y la replicación de S3 para la recuperación de desastres, el ciclo de vida de S3 para administrar los costos de almacenamiento y el bloqueo de objetos S3 para cumplir con los requisitos de cumplimiento.
- Monitoreo de su almacenamiento: el monitoreo es una parte importante a la hora de mantener la fiabilidad, la disponibilidad y el rendimiento de Amazon S3 y las soluciones de AWS. Puede monitorear la actividad y los costes de almacenamiento. Además, recomendamos que recopile los datos de monitoreo de todas las partes de la solución de AWS para que le resulte más sencillo depurar un error que se produce en distintas partes del código, en caso de que ocurra.
- Análisis e información: puede utilizar el análisis y la información en Amazon S3 a fin de comprender, analizar y optimizar el uso del almacenamiento. Por ejemplo, use Amazon S3 Storage Lens para comprender, analizar y optimizar el almacenamiento. S3 Storage Lens proporciona más de 29 métricas de uso y actividad y paneles interactivos para agregar datos para toda la organización, cuentas específicas, regiones, buckets o prefijos. Use Análisis de clases de almacenamiento para analizar los patrones de acceso al almacenamiento y decidir cuándo es el momento de mover sus datos a una clase de almacenamiento más rentable.

Desarrollar con Amazon S3

Amazon S3 es un servicio de REST. Puede enviar solicitudes a Amazon S3 con la API de REST o las bibliotecas de encapsulamiento de los SDK de AWS, que incluyen la API de REST de Amazon S3 subyacente, lo que simplifica sus tareas de programación. También puede utilizar la AWS Command Line Interface (AWS CLI) para realizar llamadas a la API de Amazon S3. Para obtener más información, consulte Realizar solicitudes.

La API de REST de Amazon S3 es una interfaz HTTP a Amazon S3. Con la API de REST, usted puede utilizar solicitudes HTTP estándar para crear, recuperar y eliminar buckets y objetos. Puede utilizar cualquier conjunto de herramientas que admita HTTP para utilizar la API de REST. Incluso puede utilizar un navegador para recuperar objetos, siempre y cuando se puedan leer de forma anónima. Para obtener más información, consulte Desarrollo con Amazon S3 mediante la API REST.

Para ayudarle a crear aplicaciones que usen el lenguaje de su elección, proporcionamos los siguientes recursos:

AWS CLI

Puede obtener acceso a las características de Amazon S3 usando AWS CLI. Para descargar y configurar AWS CLI, consulte Desarrollo con Amazon S3 mediante la AWS CLI.

La AWS CLI ofrece dos niveles de comandos para acceder a Amazon S3: comandos de alto nivel (s3) y comandos al nivel de la API (s3api y s3control). Los comandos S3 alto nivel que simplifican la realización de tareas comunes, como crear, manipular y eliminar objetos y buckets. Los comandos s3api y s3control exponen acceso directo a todas las operaciones de la API de Amazon S3, que puede utilizar para realizar operaciones avanzadas que podrían no ser posibles solamente con los comandos de alto nivel.

Para ver la lista de comandos AWS CLI de Amazon S3, consulte s3, s3api y s3control.

SDK y exploradores de AWS

Puede utilizar los SDK de AWS para desarrollar aplicaciones con Amazon S3. Los SDK de AWS simplifican las tareas de programación dado que incluyen la API de REST subyacente. Los SDK de AWS para móviles y la biblioteca de JavaScript de Amplify también están disponibles para crear con aplicaciones web y móviles conectadas usando AWS.

Además de los SDK de AWS, los exploradores de AWS están disponibles para Visual Studio y el entorno de desarrollo integrado (IDE) de Eclipse para Java. En este caso, los SDK y los exploradores están agrupados como conjuntos de herramientas de AWS.

Para obtener más información, consulte Desarrollo con Amazon S3 mediante los SDK de AWS.

Código de muestra y bibliotecas

El AWS Centro de desarrolladores y el AWS Catálogo de muestras de códigos tienen un código de muestra y bibliotecas escritas especialmente para Amazon S3. Puede utilizar estos códigos de muestra para comprender cómo implementar la API de Amazon S3. También puede ver el Referencia de la API de Amazon Simple Storage Service para comprender las operaciones de la API de Amazon S3 en detalle.

Obtener información de los tutoriales

Puede comenzar con tutoriales paso a paso para obtener información sobre Amazon S3. Están pensados para un entorno de laboratorio y usan los nombres de empresa ficticios, los nombres de

usuario y así sucesivamente. Su finalidad es proporcionar orientación general. No deben utilizarse directamente en un entorno de producción sin antes realizar una revisión y adaptación exhaustivas para satisfacer las necesidades únicas del entorno de la organización.

Introducción

- Tutorial: Almacenamiento y recuperación de un archivo con Amazon S3
- Tutorial: Introducción al uso de Amazon S3 Intelligent-Tiering
- Tutorial: Introducción al uso de las clases de almacenamiento de Amazon S3 Glacier

Optimización de costos de almacenamiento

- Tutorial: Introducción al uso de Amazon S3 Intelligent-Tiering
- Tutorial: Introducción al uso de las clases de almacenamiento de Amazon S3 Glacier
- Tutorial: Optimización de costos y aumento de visibilidad del uso con Lente de almacenamiento de S3

Administrar el almacenamiento

- Tutorial: Introducción sobre los puntos de acceso de varias regiones de Amazon S3
- Tutorial: Replicación de objetos existentes en los buckets de Amazon S3 con la replicación por lotes de S3

Alojamiento de vídeos y sitios web

- Tutorial: Alojamiento de video en streaming bajo demanda con Amazon S3, Amazon CloudFront y Amazon Route 53
- Tutorial: configuración de un sitio web estático en Amazon S3
- Tutorial: Configuración de un sitio web estático mediante un dominio personalizado registrado con Route 53

Procesamiento de datos

- Tutorial: transformación de datos para su aplicación con S3 Object Lambda
- Tutorial: detección y redacción de datos de PII con S3 Object Lambda y Amazon Comprehend

- Tutorial: Uso de S3 Object Lambda para agregar marcas de agua dinámicas a las imágenes a medida que se recuperan
- Tutorial: videos de transcodificación por lotes con operaciones por lotes de S3, AWS Lambda, y AWS Elemental MediaConvert

Protección de datos

- Tutorial: Check the Integrity of Data in Amazon S3 with Additional Checksums (Comprobación de la integridad de los datos en Amazon S3 con sumas de comprobación adicionales)
- Tutorial: Replicación de datos dentro y entre Regiones de AWS con Replicación de S3
- Tutorial: Protecting data on Amazon S3 against accidental deletion or application bugs using S3 Versioning, S3 Object Lock, and S3 Replication (Protección de los datos en Amazon S3 contra la eliminación accidental o los errores en la aplicación mediante el control de versiones de S3, S3 Object Lock y la Replicación de S3)
- Tutorial: Replicación de objetos existentes en los buckets de Amazon S3 con la replicación por lotes de S3

Explore la formación y el soporte

Usted puede aprender de AWS expertos para avanzar en sus habilidades y obtener asistencia experta para alcanzar sus objetivos.

- Formación: los recursos de formación ofrecen un enfoque práctico para conocer Amazon S3. Para obtener más información, consulte Formación y certificación deAWS y las charlas técnicas en línea de AWS.
- Foros de debate_ en el foro, puede revisar las publicaciones para saber lo que puede hacer y lo que no con Amazon S3. También puede publicar sus preguntas. Para obtener más información, consulte Foros de debate.
- Soporte técnico: si tiene más preguntas, puede ponerse en contacto con Soporte técnico.

Tutoriales

Los siguientes tutoriales presentan procedimientos integrales completos para tareas comunes Amazon S3. Están pensados para un entorno de laboratorio y usan los nombres de empresa ficticios, los nombres de usuario y así sucesivamente. Su finalidad es proporcionar orientación general. No deben utilizarse directamente en un entorno de producción sin antes realizar una revisión y adaptación exhaustivas para satisfacer las necesidades únicas del entorno de la organización.

> **Note**
>
> Para obtener más información sobre el uso de la clase de almacenamiento Amazon S3 Express One Zone con buckets de directorio, consulte ¿Qué es S3 Express One Zone? y Buckets de directorio.

Introducción

- Tutorial: Almacenamiento y recuperación de un archivo con Amazon S3
- Tutorial: Introducción al uso de Amazon S3 Intelligent-Tiering
- Tutorial: Introducción al uso de las clases de almacenamiento de Amazon S3 Glacier

Optimización de costos de almacenamiento

- Tutorial: Introducción al uso de Amazon S3 Intelligent-Tiering
- Tutorial: Introducción al uso de las clases de almacenamiento de Amazon S3 Glacier
- Tutorial: Optimización de costos y aumento de visibilidad del uso con Lente de almacenamiento de S3

Administrar el almacenamiento

- Tutorial: Introducción sobre los puntos de acceso de varias regiones de Amazon S3
- Tutorial: Replicación de objetos existentes en los buckets de Amazon S3 con la replicación por lotes de S3

Alojamiento de vídeos y sitios web

- Tutorial: Alojamiento de video en streaming bajo demanda con Amazon S3, Amazon CloudFront y Amazon Route 53
- Tutorial: configuración de un sitio web estático en Amazon S3
- Tutorial: Configuración de un sitio web estático mediante un dominio personalizado registrado con Route 53

Procesamiento de datos

- Tutorial: transformación de datos para su aplicación con S3 Object Lambda
- Tutorial: detección y redacción de datos de PII con S3 Object Lambda y Amazon Comprehend
- Tutorial: Uso de S3 Object Lambda para agregar marcas de agua dinámicas a las imágenes a medida que se recuperan
- Tutorial: videos de transcodificación por lotes con operaciones por lotes de S3, AWS Lambda, y AWS Elemental MediaConvert

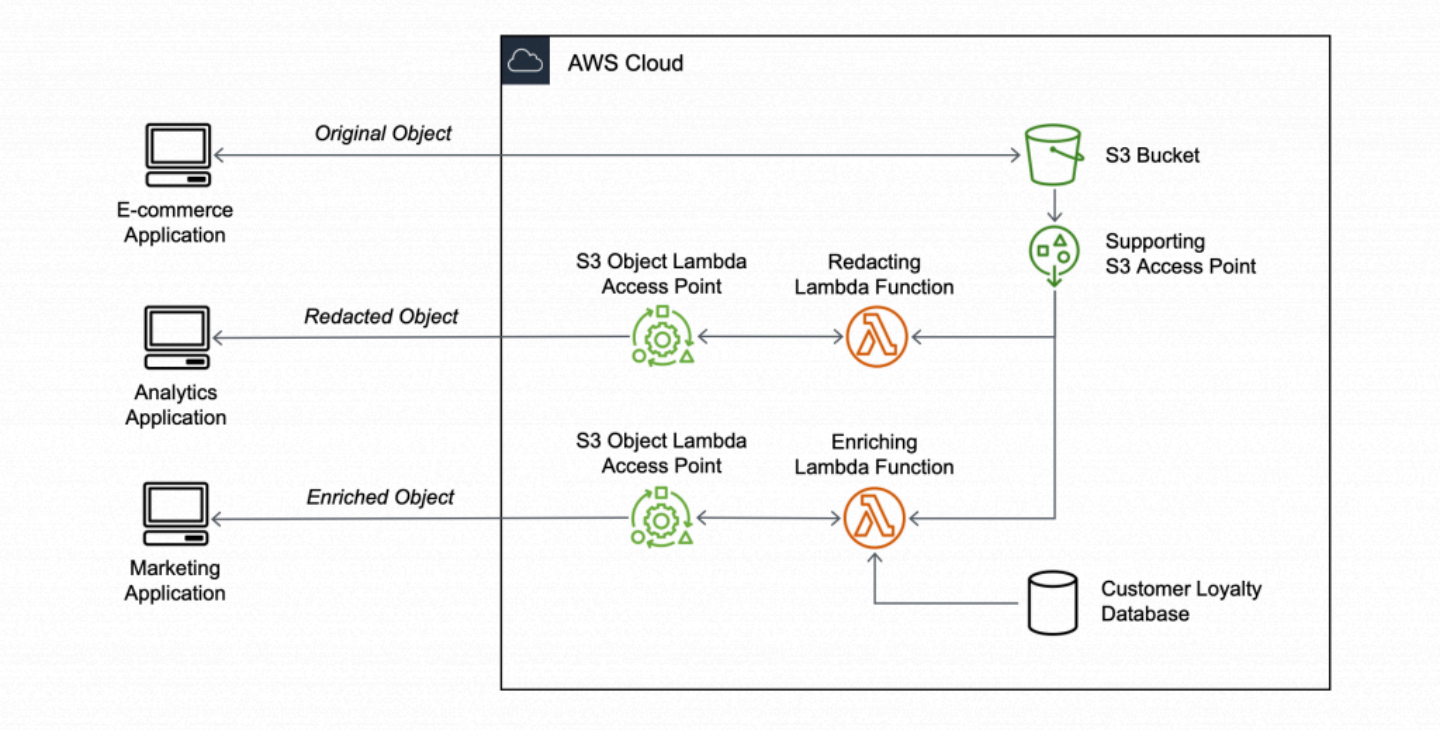
Protección de datos

- Tutorial: Check the Integrity of Data in Amazon S3 with Additional Checksums (Comprobación de la integridad de los datos en Amazon S3 con sumas de comprobación adicionales)
- Tutorial: Replicación de datos dentro y entre Regiones de AWS con Replicación de S3
- Tutorial: Protecting data on Amazon S3 against accidental deletion or application bugs using S3 Versioning, S3 Object Lock, and S3 Replication (Protección de los datos en Amazon S3 contra la eliminación accidental o los errores en la aplicación mediante el control de versiones de S3, S3 Object Lock y la Replicación de S3)
- Tutorial: Replicación de objetos existentes en los buckets de Amazon S3 con la replicación por lotes de S3

Tutorial: transformación de datos para su aplicación con S3 Object Lambda

Cuando almacena datos en Amazon S3, puede compartirlos fácilmente para utilizarlos en varias aplicaciones. Sin embargo, cada aplicación puede tener requisitos únicos de formato de datos y es posible que necesite modificar o procesar sus datos para un caso de uso específico. Por ejemplo, un conjunto de datos creado por una aplicación de comercio electrónico puede incluir información de identificación personal (PII). Cuando se procesan los mismos datos para análisis, esta PII no es necesaria y debe ser borrada. Sin embargo, si se utiliza el mismo conjunto de datos para una campaña de marketing, es posible que deba enriquecer los datos con detalles adicionales, como información de la base de datos de fidelización del cliente.

Con S3 Object Lambda, puede agregar su propio código para procesar los datos recuperados de S3 antes de devolverlos a una aplicación. En concreto, puede configurar una función AWS Lambda y asociarla a un punto de acceso de S3 Object Lambda. Cuando una aplicación envía solicitudes estándar S3 a través del punto de acceso de S3 Object Lambda, se invoca la función de Lambda especificada para procesar cualquier dato recuperado de un bucket de S3 a través del punto de acceso de S3 compatible. Luego, el punto de acceso de S3 Object Lambda devuelve el resultado transformado a la aplicación. Puede crear y ejecutar sus propias funciones Lambda personalizadas, adaptando la transformación de datos de S3 Object Lambda a su caso de uso específico, todo ello sin necesidad de cambios en suAplicaciones de.



Objetivo

En este aprendizaje, aprenderá a agregar código personalizado a las solicitudes GET S3 estándar para modificar el objeto solicitado recuperado de S3 de modo que el objeto se ajuste a las necesidades del cliente o aplicación solicitante. Específicamente, aprenderá cómo transformar todo el texto del objeto original almacenado en S3 a mayúsculas a través de S3 Object Lambda.

> **Note**
>
> En este tutorial, se usa código Python para transformar los datos. Para ver ejemplos sobre cómo usar otros AWS SDK, consulte Transformación de datos para su aplicación con S3 Object Lambda en la biblioteca de ejemplos de código de AWS SDK.

Temas

Requisitos previos

Antes de empezar este tutorial, debe tener una Cuenta de AWS en la que puede iniciar sesión como usuario de AWS Identity and Access Management (IAM) con los permisos correctos. También debe instalar la versión 3.8 o posterior de Python.

Pasos secundarios

- Crear un usuario de IAM con permisos en la Cuenta de AWS (consola)
- Instale Python 3.8 o posterior en su equipo local

Crear un usuario de IAM con permisos en la Cuenta de AWS (consola)

Puede crear un usuario de IAM para el tutorial. Para completar este tutorial, el usuario de IAM debe adjuntar las siguientes políticas de IAM para acceder a los recursos de AWS y realizar acciones específicas. Para obtener más información acerca de cómo crear un usuario de IAM, consulte Creación del primer grupo de usuarios y administradores de IAM en la guía del usuario de IAM.

El usuario de IAM requiere las siguientes directivas:

- AmazonS3FullAccess: otorga permisos a todas las acciones de Amazon S3, incluidos los permisos para crear y utilizar un punto de acceso de Object Lambda.
- AWSLambda_FullAccess— Otorga permisos a todas las acciones de Lambda.
- IAMFullAccess— Otorga permisos a todas las acciones de IAM.
- IAMAccessAnalyzerReadOnlyAccess: concede permisos para leer toda la información de acceso proporcionada por IAM Access Analyzer.
- CloudWatchLogsFullAccess: concede acceso total a Registros de CloudWatch.

> **Note**
>
> Para simplificar, este tutorial crea y utiliza un usuario de IAM. Después de completar este tutorial, recuerde Eliminación del rol de IAM. Para uso en producción, le recomendamos que siga las Prácticas recomendadas de seguridad de IAM en la Guía del usuario de IAM. Una práctica recomendada exige que los usuarios humanos utilicen la federación con un proveedor de identidades para acceder a AWS con credenciales temporales. Otra práctica recomendada es exigir a las cargas de trabajo que utilicen credenciales temporales con roles de IAM para acceder a AWS. Para obtener información sobre el uso de AWS IAM Identity Center para crear usuarios con credenciales temporales, consulte Getting started (Introducción) en la Guía del usuario de AWS IAM Identity Center.
> Este tutorial también utiliza políticas administradas de AWS de acceso completo. Para uso en producción, le recomendamos que otorgue solo los permisos mínimos necesarios para su caso de uso, de acuerdo con las prácticas recomendadas de seguridad.

Instale Python 3.8 o posterior en su equipo local

Utilice el siguiente procedimiento para instalar Python 3.8 o posterior en su equipo local. Para obtener instrucciones de instalación, consulte la página Downloading Python en la Guía para principiantes de Python.

1. Abra su terminal o shell local y ejecute el siguiente comando para determinar si Python ya está instalado y, en caso afirmativo, qué versión está instalada.

   ```
   python --version
   ```

2. Si no dispone de Python 3.8 ni posterior, descargue el Instalador oficial de Python 3.8 o posterior que sea adecuado para su máquina local.
3. Ejecute el instalador haciendo doble clic en el archivo descargado y siga los pasos para completar la instalación.

   Para Usuarios de Windows, elija Agregar Python 3.X a PATH en el asistente de instalación antes de elegir Instalar ahora.
4. Reinicie el terminal cerrándolo y volviéndolo a abrirlo.
5. Ejecute el siguiente comando para verificar que Python 3.8 o posterior se instaló correctamente.

   Para Usuarios de macOS, ejecute este comando:

   ```
   python3 --version
   ```

   Para usuarios de Windows, ejecute este comando:

   ```
   python --version
   ```

6. Ejecute el siguiente comando para comprobar que el administrador de paquetes pip3 está instalado. Si ve un número de versión pip y python 3.8 o posterior en la respuesta del comando, eso significa que el gestor de paquetes pip3 se instaló correctamente.

   ```
   pip --version
   ```

Paso 1: Crear un bucket de S3

Cree un bucket para almacenar los datos originales que tiene previsto transformar.

Creación de un bucket

1. Inicie sesión AWS Management Console Management Console y abra la consola de Amazon S3 en https://console.aws.amazon.com/s3/.
2. En el panel de navegación izquierdo, elija Instancias.
3. Elija Crear bucket.

   Se abrirá la página Crear bucket.
4. En Nombre de bucket, escriba el nombre del bucket (por ejemplo: **`tutorial-bucket`**).

   Para obtener más información acerca de las reglas de nomenclatura del bucket de Amazon S3, consulte Reglas de nomenclatura de buckets.
5. En Región, elija la Región de AWS en la que desea que se encuentre el bucket.

   Para obtener más información acerca de bucket Region, consulte Descripción general de los buckets.
6. Para Configuración de Block Public Access para este bucket, conserve la configuración predeterminada (Bloquear todo acceso público está habilitado).

   Le recomendamos que deje todas las configuraciones habilitadas a menos que sepa que necesita desactivar una o varias de ellas para su caso de uso, como alojar un sitio web público. Para obtener más información acerca del bloqueo del acceso público, consulte Bloquear el acceso público a su almacenamiento de Amazon S3.
7. Mantenga la configuración restante establecida en los valores predeterminados.

   (Opcional) Si desea configurar opciones de bucket adicionales para el caso de uso específico, consulteCrear un bucket.
8. Elija Crear bucket.

Paso 2: cargar un archivo al bucket de S3.

Cargue el archivo en un bucket de &S3;. Este archivo de texto contiene los datos originales que transformará a mayúsculas más adelante en este tutorial.

Por ejemplo, puede cargar un archivo `tutorial.txt` que contiene el siguiente texto:

```
Amazon S3 Object Lambda Tutorial:
You can add your own code to process data retrieved from S3 before
```

```
returning it to an application.
```

Para cargar un archivo en un bucket

1. Inicie sesión AWS Management Console Management Console y abra la consola de Amazon S3 en https://console.aws.amazon.com/s3/.
2. En el panel de navegación izquierdo, elija Instancias.
3. En la lista Buckets, elija el nombre del bucket que creó en elPaso 1 (por ejemplo: **`tutorial-bucket`**) para cargar el archivo.
4. En la pestaña Objetos del bucket, elija Cargar.
5. En la página Cargar, en Archivos y carpetas, elija Añadir archivos.
6. Seleccione un archivo que cargar y luego seleccione Abrir. Por ejemplo, puede cargar el archivo de ejemplo `tutorial.txt` mencionado anteriormente.
7. Seleccione Cargar.

Paso 3: Crear un punto de acceso de S3

Para utilizar un punto de acceso de S3 Object Lambda para acceder y transformar los datos originales, debe crear un punto de acceso de S3 y asociarlo con el bucket de S3 que creó en el Paso 1. El punto de acceso debe estar en la misma Región de AWS que los objetos que desea transformar.

Más adelante en este tutorial, utilizará este punto de acceso como punto de acceso de soporte para su punto de acceso de Object Lambda.

Para crear un punto de acceso

1. Inicie sesión AWS Management Console Management Console y abra la consola de Amazon S3 en https://console.aws.amazon.com/s3/.
2. En el panel de navegación, elija Puntos de acceso.
3. En la página de Puntos de acceso, elija Crear punto de acceso.
4. Introduzca el nombre que desee (por ejemplo: **`tutorial-access-point`**) para el punto de acceso en el campo Nombre del punto de acceso.

   Para obtener más información sobre cómo asignar nombres a los puntos de acceso, consulte Reglas para asignar nombres a los puntos de acceso de Amazon S3.

5. En el campo Nombre del Bucket escriba el nombre del bucket que creó en el Paso 1 (por ejemplo: **`tutorial-bucket`**). A continuación, asocie dicha política al bucket de S3.

   (Opcional) Puede elegir Examinar S3) para explorar su cuenta y buscar buckets. Si elige Examinar S3, seleccione el bucket que le interese y seleccione Elegir ruta para rellenar el campo Nombre del bucket con el nombre del bucket en cuestión.

6. Para Origen de la red, elija Internet.

   Para obtener más información acerca de los orígenes de red para los puntos de acceso, consulte Crear puntos de acceso restringidos a una nube privada virtual.

7. Todas las configuraciones de bloqueo de acceso público están habilitadas de forma predeterminada para los puntos de acceso. Le recomendamos que deje habilitada Block all public access (Bloquear todo el acceso público).

   Para obtener más información, consulte Administrar el acceso público a los puntos de acceso.

8. En el resto de configuraciones de punto de acceso, mantenga la configuración predeterminada.

   (Opcional) Puede modificar la configuración del punto de acceso para admitir el caso de uso. En este tutorial, le recomendamos que mantenga la configuración predeterminada.

   (Opcional) Si necesita administrar el acceso al punto de acceso, puede especificar una directiva de punto de acceso. Para obtener más información, consulte Ejemplos de políticas de puntos de acceso.

9. Elija Crear punto de acceso.

Paso 4: Crear una función de Lambda

Para transformar los datos originales, cree una función de Lambda para utilizarla con el punto de acceso de S3 Object Lambda.

Pasos secundarios

- Escribir el código de función de Lambda y crear un paquete de implementación con un entorno virtual
- Crear una función de Lambda con un rol de ejecución (consola)
- Implemente su código de función de Lambda con archivos .zip y configure la función de Lambda (consola)

Escribir el código de función de Lambda y crear un paquete de implementación con un entorno virtual

1. En el equipo local, cree una carpeta con el nombre de carpeta`object-lambda`Para utilizarlo más adelante en este tutorial del entorno virtual.
2. En la carpeta `object-lambda`, cree un archivo con una función de Lambda que cambie todo el texto del objeto original a mayúsculas. Por ejemplo, puede utilizar la siguiente función escrita en Python. Guarde esta función en un archivo llamado`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**
>
> La función de Lambda de ejemplo anterior carga todo el objeto solicitado en la memoria antes de transformarlo y devolverlo al cliente. Alternativamente, puede transmitir el objeto desde S3 para evitar cargar todo el objeto en la memoria. Este enfoque puede resultar útil cuando se trabaja con objetos grandes. Para obtener más información acerca del streaming de respuestas con puntos de acceso Object Lambda, consulte los ejemplos de transmisión en Trabajar con solicitudes `GetObject` en Lambda.

Cuando está escribiendo una función de Lambda para su uso con un punto de acceso de S3 Object Lambda, la función se basa en el contexto de evento de entrada que S3 Object Lambda proporciona a la función de Lambda. S3 Object Lambda proporciona contexto sobre la solicitud que se realiza en el evento pasado a Lambda. Contiene los parámetros que utiliza para crear la función de Lambda.

Los campos utilizados para crear la función de Lambda anterior son los siguientes:

El campo de `getObjectContext` se refiere a los detalles de entrada y salida de las conexiones a Amazon S3 y S3 Object Lambda. Tiene los subcampos siguientes:

- `inputS3Url`— URL prefirmada que la función de Lambda puede utilizar para descargar el objeto original desde el punto de acceso de soporte. Al utilizar una URL prefirmada, la función de Lambda no necesita tener permisos de lectura de Amazon S3 para recuperar el objeto original y solo puede acceder al objeto procesado por cada invocación.
- `outputRoute` - un token de enrutamiento que se agrega a la URL de S3 Object Lambda cuando la función de Lambda llama a `WriteGetObjectResponse`.
- `outputToken`: un token utilizado por S3 Object Lambda para hacer coincidir el`WriteGetObjectResponse`con la persona que llama original al enviar de vuelta el objeto transformado.

Para obtener más información acerca de todos los campos del contexto de evento, consulte Formato y uso del contexto del evento y Escritura de funciones de Lambda para puntos de acceso de S3 Object Lambda.

3. En el terminal local, introduzca el siguiente comando para instalar el paquete `virtualenv`:

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

4. En su terminal local, abra el `object-lambda` que creó anteriormente y, a continuación, escriba el siguiente comando para crear e inicializar un entorno virtual denominado `venv`.

```
python -m virtualenv venv
```

5. Para activar el entorno virtual, ingrese el siguiente comando para ejecutar el archivo `activate` desde la carpeta del entorno:

   Para Usuarios de macOS, ejecute este comando:

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

   En Windows, ejecute este comando:

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

   El símbolo del sistema cambia para mostrar (venv) que indica que el entorno virtual está activo.

6. Para instalar las bibliotecas requeridas, ejecute los siguientes comandos línea por línea en el `venv` entorno virtual.

   Estos comandos instalan versiones actualizadas de las dependencias de su `lambda_handler` función de Lambda. Estas dependencias son AWS SDK for Python (Boto3) y el módulo de solicitudes.

```
pip3 install boto3
```

```
pip3 install requests
```

7. Para desactivar el entorno virtual, puede ejecutar el siguiente comando:

```
deactivate
```

8. Para crear un paquete de implementación con las bibliotecas instaladas como un archivo `.zip` llamado `lambda.zip` en la raíz del directorio `object-lambda`, ejecute los siguientes comandos línea por línea en el terminal local.

> **Tip**
>
> Es posible que sea necesario ajustar los siguientes comandos para que funcionen en su entorno concreto. Por ejemplo, una biblioteca puede aparecer en `site-packages` o `dist-packages`, y la primera carpeta podría ser `lib` o `lib64`. Además, la carpeta `python` puede tener un nombre con una versión de Python diferente. Puede utilizar el comando `pip show` para localizar un paquete específico.

Para Usuarios de macOS ejecute estos comandos:

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

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

Para Usuarios de Windows ejecute estos comandos:

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

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

El último comando guarda el paquete de implementación en la raíz del directorio `object-lambda`.

9. Agregue archivos de código de función `transform.py` a la raíz del paquete de implementación.

   Para Usuarios de macOS ejecute estos comandos:

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

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

   Para Usuarios de Windows ejecute estos comandos:

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

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

Cuando realice este paso, tendrá la siguiente estructura de directorio:

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

Crear una función de Lambda con un rol de ejecución (consola)

1. Inicie sesión en la AWS Management Console y abra la consola AWS Lambda en https://console.aws.amazon.com/lambda/.

2. En el panel de navegación izquierdo, elija Instancias.
3. Elija Create function (Crear función).
4. Elija Author from scratch (Crear desde cero).
5. Bajo Basic information (Información básica), haga lo siguiente:

   a. En Function name (Nombre de función), introduzca **`tutorial-object-lambda-function`**.

   b. En Runtime (Tiempo de ejecución):, elija Python 3.8 o una versión posterior.

6. Expanda la sección Change default execution role (Cambiar el rol de ejecución predeterminado). En Execution role (Rol de ejecución), elija Create a new role with basic Lambda permissions (Crear un nuevo rol con permisos básicos de Lambda).

   En el Paso 5 más adelante en este tutorial, adjunte el archivoAmazonS3ObjectLambdaExecutionRolePolicy a este rol de ejecución de la función de Lambda.

7. Mantenga la configuración restante establecida en los valores predeterminados.
8. Elija Crear función.

Implemente su código de función de Lambda con archivos .zip y configure la función de Lambda (consola)

1. En la consola AWS Lambda en https://console.aws.amazon.com/lambda/, elija Funciones En el panel de navegación izquierdo.
2. Elija la función de Lambda que creó anteriormente (por ejemplo, **`tutorial-object-lambda-function`**).
3. En la página de detalles de la función de Lambda, elija la pestaña Code (Código). En el sector Code Source (Código fuente), seleccione Upload from (Cargar desde) y luego .zip file (archivo .zip).
4. Seleccione Upload (Cargar) para seleccionar el archivo `.zip` local.
5. Elija el archivo `lambda.zip` que creó anteriormente y, a continuación, elija Open (Abrir).
6. Seleccione Save (Guardar).
7. En la sección Runtime settings (Configuración de tiempo de ejecución), elija Edit (Editar).
8. En la página Edit runtime settings (Editar la configuración de tiempo de ejecución), confirme que Runtime (Tiempo de ejecución): toma el valor Python 3.8 o una versión posterior.
9. Para decirle al tiempo de ejecución de Lambda qué método de controlador en su código de función de Lambda invocar, ingrese **`transform.lambda_handler`**: paraHandler (Controlador):.

   Al configurar una función en Python, el valor del controlador es el nombre del archivo y el nombre de un módulo del controlador exportado, separados por un punto. Por ejemplo, `transform.lambda_handler` llama al método `lambda_handler` definido en `transform.py`.
10. Seleccione Guardar.
11. (Opcional) En la página de detalles de la función de Lambda, seleccione la pestaña Configuration (Configuración). En el panel de navegación izquierdo, elija General configuration (Configuración general) y, a continuación, elija Edit (Editar). En el campo Timeout (Tiempo de espera), introduzca **`1`** min **`0`** seg. Mantenga la configuración restante establecida en los valores predeterminados y elija Save (Guardar).

    Timeout (Tiempo de espera): período durante el cual Lambda permite que se ejecute una función antes de pararla. El valor predeterminado es de 3 segundos. La duración máxima de una función de Lambda utilizada por S3 Object Lambda es de 60 segundos. Los precios se basan

en la cantidad de memoria configurada y en la cantidad de tiempo durante la que se ejecuta el código.

Paso 5: Configurar una política de IAM para el rol de ejecución de su función de Lambda

Para habilitar la función de Lambda para proporcionar datos personalizados y encabezados de respuesta a `GetObject`, el rol de ejecución de la función de Lambda debe tener permisos de IAM para llamar a la API de `WriteGetObjectResponse`.

Para asociar una política de IAM a su función de Lambda

1. En la consola AWS Lambda en https://console.aws.amazon.com/lambda/, elija Functions (Funciones) en el panel de navegación izquierdo.
2. Elija la función que creó en el paso 4 (por ejemplo: **`tutorial-object-lambda-function`**).
3. En la página de detalles de su función de Lambda, elija la opción Configuration (Configuración) y, a continuación, elija Permissions (Permisos) en el panel de navegación izquierdo.
4. En Execution role (Rol de ejecución), elija el enlace del Role name (Nombre del rol). Abra la consola de IAM.
5. En la página Summary (Resumen) de la consola de IAM para el rol de ejecución de la función de Lambda, elija la pestaña Permissions (Permisos). A continuación, en el menú Add Permissions (Agregar permisos), elija Attach policies (Asociar políticas).
6. En la página Attach Permissions (Asociar permisos), introduzca **`AmazonS3ObjectLambdaExecutionRolePolicy`** en el campo de búsqueda para filtrar la lista de políticas. Active la casilla de verificación que hay junto al nombre del cuadro de diálogoAmazonS3ObjectLambdaExecutionRolePolicypolítica.
7. Seleccione Asociar políticas.

Paso 6: Crear un punto de acceso de S3 Object Lambda

Un punto de acceso de S3 Object Lambda proporciona la flexibilidad para invocar una función de Lambda directamente desde una solicitud GET S3 para que la función pueda procesar datos recuperados de un punto de acceso de S3. Al crear y configurar un punto de acceso de S3 Object Lambda, debe especificar la función de Lambda para invocar y proporcionar el contexto de evento en formato JSON como parámetros personalizados para utilizar Lambda.

Para crear un punto de acceso de S3 Object Lambda

1. Inicie sesión AWS Management Console Management Console y abra la consola de Amazon S3 en https://console.aws.amazon.com/s3/.
2. En el panel de navegación del lado izquierdo de la consola, elija Object Lambda Access Points (Puntos de acceso de Object Lambda).
3. En la página Object Lambda Access Points (Puntos de acceso Object Lambda), elija Create Object Lambda Access Point (Crear un punto de acceso de Object Lambda).
4. En Nombre del punto de acceso del objeto Lambda, introduzca el nombre que desea utilizar para el punto de acceso del objeto Lambda (por ejemplo: **`tutorial-object-lambda-accesspoint`**).
5. Para Supporting Access Point (Soporte de punto de acceso), introduzca o busque el punto de acceso estándar que ha creado en el Paso 3(por ejemplo: **`tutorial-access-point`**), y luego elija Choose Supporting Access Point (Elegir soporte de punto de acceso).
6. Para las S3 APIs (API de S3), para recuperar objetos del bucket de S3 para que la función de Lambda los procese, seleccione GetObject.
7. Para Invoke Lambda function (Invocación de una función de Lambda), puede elegir una de las dos opciones siguientes.
   - Elegir Choose from functions in your account (Elija entre las funciones de su cuenta) y, a continuación, elija la función de Lambda que ha creado en el Paso 4 (por ejemplo: **`tutorial-object-lambda-function`**) desde la lista desplegable Lambda function (Función de Lambda).
   - Elegir Enter ARN (Ingresar ARN) y, a continuación, introduzca el nombre de recurso de Amazon (ARN) de la función de Lambda que ha creado en el Paso 4.
8. Para Lambda function version (Versión de función de Lambda), elija $LATEST (la última versión de la función de Lambda que ha creado en el Paso 4).
9. (Opcional) Si necesita su función de Lambda para reconocer y procesar solicitudes GET con encabezados de rango y número de pieza, seleccione Lambda function supports requests using range (La función de Lambda admite solicitudes usando rango) y Lambda function supports requests using part numbers (La función de Lambda admite solicitudes usando números de pieza). De lo contrario, desactive estas dos casillas de verificación.

   Para obtener más información acerca de cómo utilizar números de rango o rango con S3 Object Lambda, consulte Trabajar con encabezados Range y partNumber.

10. (Opcional) En Payload - optional (Carga - opcional), agregue texto JSON para proporcionar información adicional a su función de Lambda.

    Una carga es texto JSON opcional que puede proporcionar a su función de Lambda como entrada para todas las invocaciones procedentes de un punto de acceso de S3 Object Lambda específico. Puede configurar cargas con diferentes parámetros para diferentes puntos de acceso Object Lambda que invoquen la misma función de Lambda, ampliando así la flexibilidad de su función de Lambda.

    Para obtener más información acerca de patones de rutas, consulte Formato y uso del contexto del evento.
11. (Opcional) Para las Métricas de solicitudes - opcional, elija Deshabilitar o Habilitar para agregar la supervisión de Amazon S3 al punto de acceso del objeto Lambda. Las métricas de solicitud se facturan según la tarifa de Amazon CloudWatch estándar. Para obtener más información, consulte los precios de CloudWatch.
12. En Object Lambda Access Point policy - optional (Política de punto de acceso Object Lambda - opcional), conserve la configuración predeterminada.

    (Opcional) Ejecute para establecer la política de recursos. Esta política de recursos concede el permiso de la API `GetObject` para usar el punto de acceso del objeto Lambda específico.
13. Mantenga la configuración restante establecida en los valores predeterminados y elija Create Object Lambda Access Point (Creación de punto de acceso Object Lambda).

Paso 7: Ver los datos transformados

Ahora, S3 Object Lambda está listo para transformar sus datos para su caso de uso. En este tutorial, S3 Object Lambda transforma todo el texto de su objeto a mayúsculas.

Pasos secundarios

- Visualización de los datos transformados en el punto de acceso de S3 Object Lambda
- Ejecución de una secuencia de comandos de Python para imprimir los datos originales y transformados

Visualización de los datos transformados en el punto de acceso de S3 Object Lambda

Cuando solicita recuperar un archivo a través de su punto de acceso de S3 Object Lambda, cree una llamada de la API `GetObject` a S3 Object Lambda. S3 Object Lambda invoca la función de

Lambda para transformar sus datos, y luego devuelve los datos transformados como la respuesta a la llamada a la API estándar S3 `GetObject`.

1. Inicie sesión AWS Management Console Management Console y abra la consola de Amazon S3 en https://console.aws.amazon.com/s3/.
2. En el panel de navegación, elija Object Lambda Access Points (Puntos de acceso de Object Lambda).
3. En la página Puntos de acceso del objeto lambda, elija el punto de acceso de S3 Object Lambda que ha creado en el Paso 6 (por ejemplo, **`tutorial-object-lambda-accesspoint`**).
4. En la pestaña Objetos de su punto de acceso de S3 Object Lambda, seleccione el archivo que tenga el mismo nombre (por ejemplo, `tutorial.txt`) como el que cargó en el bucket de S3 en el Paso 2.

   Este archivo debe contener todos los datos transformados.
5. Para ver los datos transformados, elija Open (Abrir) o Download (Descargar).

Ejecución de una secuencia de comandos de Python para imprimir los datos originales y transformados

Puede utilizar S3 Object Lambda con sus aplicaciones existentes. Para ello, actualice la configuración de la aplicación para utilizar el nuevo ARN de punto de acceso de S3 Object Lambda que creó en el Paso 6 para recuperar datos desde S3.

El siguiente ejemplo de secuencia de comandos de Python imprime tanto los datos originales del bucket de S3 como los datos transformados desde el punto de acceso de S3 Object Lambda.

1. Inicie sesión AWS Management Console Management Console y abra la consola de Amazon S3 en https://console.aws.amazon.com/s3/.
2. En el panel de navegación, elija Object Lambda Access Points (Puntos de acceso de Object Lambda).
3. En la página Puntos de acceso del objeto Lambda, elija el botón de opción situado a la izquierda del punto de acceso de S3 Object Lambda que ha creado en el Paso 6(por ejemplo, **`tutorial-object-lambda-accesspoint`**).
4. Seleccionar Copy ARN (Copiar ARN).
5. Guarde el ARN para utilizarlo más tarde.

6. Escriba un script de Python en su máquina local para imprimir los datos originales (por ejemplo, `tutorial.txt`) de su S3 Bucket y los datos transformados (por ejemplo, `tutorial.txt`) desde el punto de acceso de S3 Object Lambda). Puede realizar una prueba con el siguiente script de ejemplo.

```
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. Guarde su script de Python con un nombre personalizado (por ejemplo: `tutorial_print.py`) en la carpeta (por ejemplo: `object-lambda`) que ha creado en el Paso 4 en el equipo local.
8. En el terminal local, ejecute el siguiente comando desde la raíz del directorio (por ejemplo: `object-lambda`) que ha creado en el Paso 4.

```
python3 tutorial_print.py
```

Debería ver tanto los datos originales como los datos transformados (todo el texto en mayúsculas) a través del terminal. Debería ver algo parecido a lo siguiente.

```
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.
```

Paso 8: Eliminación

Si transformó sus datos a través de S3 Object Lambda solo como un ejercicio de aprendizaje, elimine los recursos de AWS que asignó para dejar de acumular cargos.

Pasos secundarios

- Eliminar el punto de acceso del objeto Lambda
- Elimine el punto de acceso de S3
- Busque el rol de ejecución de la función de Lambda.
- Para eliminar la función de Lambda
- Eliminación del grupo de registros de CloudWatch
- Elimine el archivo original en el bucket de origen de S3
- Eliminar el bucket de origen de S3
- Eliminación del rol de IAM

Eliminar el punto de acceso del objeto Lambda

1. Inicie sesión AWS Management Console Management Console y abra la consola de Amazon S3 en https://console.aws.amazon.com/s3/.
2. En el panel de navegación, elija Object Lambda Access Points (Puntos de acceso de Object Lambda).
3. En la página Puntos de acceso del objeto Lambda, elija el botón de opción situado a la izquierda del punto de acceso de S3 Object Lambda que ha creado en el Paso 6 (por ejemplo, **`tutorial-object-lambda-accesspoint`**).

4. Elija Delete (Eliminar).
5. Confirme que desea eliminar el punto de acceso del objeto Lambda. Para ello, escriba su nombre en el campo de texto que aparece y elija Eliminar.

Elimine el punto de acceso de S3

1. Inicie sesión AWS Management Console Management Console y abra la consola de Amazon S3 en https://console.aws.amazon.com/s3/.
2. En el panel de navegación, elija Access Points (Puntos de acceso).
3. Desplácese hasta el punto de acceso que creó en el Paso 3 (por ejemplo: **`tutorial-access-point`**) y elija el botón de opción situado junto al nombre del punto de acceso.
4. Elija Eliminar.
5. Confirme que desea eliminar el punto de acceso escribiendo su nombre en el campo de texto que aparece y elija Delete (Eliminar).

Busque el rol de ejecución de la función de Lambda.

1. Inicie sesión en la AWS Management Console y abra la consola AWS Lambda en https://console.aws.amazon.com/lambda/.
2. En el panel de navegación izquierdo, elija Instancias.
3. Elija la función que creó en el paso 4 (por ejemplo: **`tutorial-object-lambda-function`**).
4. En la página de detalles de su función de Lambda, elija la opción Configuration (Configuración) y, a continuación, elija Permissions (Permisos) en el panel de navegación izquierdo.
5. En Execution role (Rol de ejecución), elija el enlace del Role name (Nombre del rol). Abra la consola de IAM.
6. En la página Summary (Resumen) de la consola de IAM de la página de ejecución de su función de Lambda, elija Delete role (Eliminar rol).
7. En el cuadro de diálogo Delete role (Eliminar rol), elija Yes, delete (Sí, eliminar).

Para eliminar la función de Lambda

1. En la consola AWS Lambda en https://console.aws.amazon.com/lambda/, elija Funciones en el panel de navegación izquierdo.

2. Active la casilla de verificación situada a la izquierda del nombre de la función que ha creado en el Paso 4 (por ejemplo: **`tutorial-object-lambda-function`**).
3. Elija Acciones y, a continuación, elija Eliminar.
4. En el cuadro de diálogo Eliminar función, elija Eliminar.

Eliminación del grupo de registros de CloudWatch

1. Abra la consola de CloudWatch en https://console.aws.amazon.com/cloudwatch/.
2. En el panel de navegación izquierdo, elija Logs (Registros), Log groups (Grupos de registros).
3. Busque el grupo de registros cuyo nombre termina con la función de Lambda que creó en el Paso 4(por ejemplo: **`tutorial-object-lambda-function`**).
4. Active la casilla de verificación situada a la izquierda del nombre del grupo de registros.
5. Elija Actions (Acciones) y, a continuación, elija Delete log group (Eliminar grupo de registro).
6. En el cuadro de diálogo Delete log group(s), Eliminar grupo(s) de registro(s) elija Delete (Eliminar).

Elimine el archivo original en el bucket de origen de S3

1. Inicie sesión AWS Management Console Management Console y abra la consola de Amazon S3 en https://console.aws.amazon.com/s3/.
2. En el panel de navegación izquierdo, elija Instancias.
3. En la lista Bucket name (Nombre del Bucket), seleccione el nombre del bucket al que ha subido el archivo original en el Paso 2 (por ejemplo: **`tutorial-bucket`**).
4. Seleccione la casilla de verificación situada a la izquierda de los nombres de los objetos que desea eliminar (por ejemplo: `tutorial.txt`).
5. Elija Eliminar.
6. En la página Delete objects (Eliminar objetos), en la sección Permanently delete objects? (¿Eliminar objetos de forma permanente?), confirme que desea eliminar este objeto escribiendo **`permanently delete`** en el cuadro de texto.
7. Elija Eliminar objetos.

Eliminar el bucket de origen de S3

1. Inicie sesión AWS Management Console Management Console y abra la consola de Amazon S3 en https://console.aws.amazon.com/s3/.
2. En el panel de navegación izquierdo, elija Instancias.
3. En la lista Buckets, elija el botón de opción situado junto al nombre del bucket que ha creado en el Paso 1 (por ejemplo: **`tutorial-bucket`**).
4. Elija Eliminar.
5. En la página Delete bucket (Eliminar bucket) confirme que desea eliminar el bucket introduciendo el nombre del bucket en el campo de texto y, a continuación, elija Delete bucket (Eliminar bucket).

Eliminación del rol de IAM

1. Inicie sesión en AWS Management Console Management Console y abra la consola IAM en https://console.aws.amazon.com/iam/.
2. En el panel de navegación, elija Roles y, a continuación, seleccione la casilla de verificación junto al nombre del rol que desee eliminar.
3. En la parte superior de la página, elija Delete user (Eliminar usuario).
4. En el cuadro de diálogo Delete ***`user name`***? (¿Eliminar Nombre de usuario?), introduzca el nombre de usuario en el campo de entrada de texto para confirmar la eliminación del usuario. Elija Eliminar.

Siguientes pasos

Después de completar este tutorial, puede personalizar la función de Lambda para su caso de uso para modificar los datos devueltos por solicitudes GET de S3 estándar.

La siguiente es una lista de casos de uso comunes para S3 Object Lambda:

- Enmascarar datos confidenciales para garantizar la seguridad y el cumplimiento.

  Para obtener más información, consulte Tutorial: detección y redacción de datos de PII con S3 Object Lambda y Amazon Comprehend.
- Filtrar ciertas filas de datos para entregar información específica.

- Aumentar los datos con información de otros servicios o bases de datos.
- Convertir entre formatos de datos, como convertir XML a JSON para la compatibilidad de aplicaciones.
- Comprimir o descomprimir archivos a medida que se descargan.
- Cambio de tamaño y marcas de agua de las imágenes

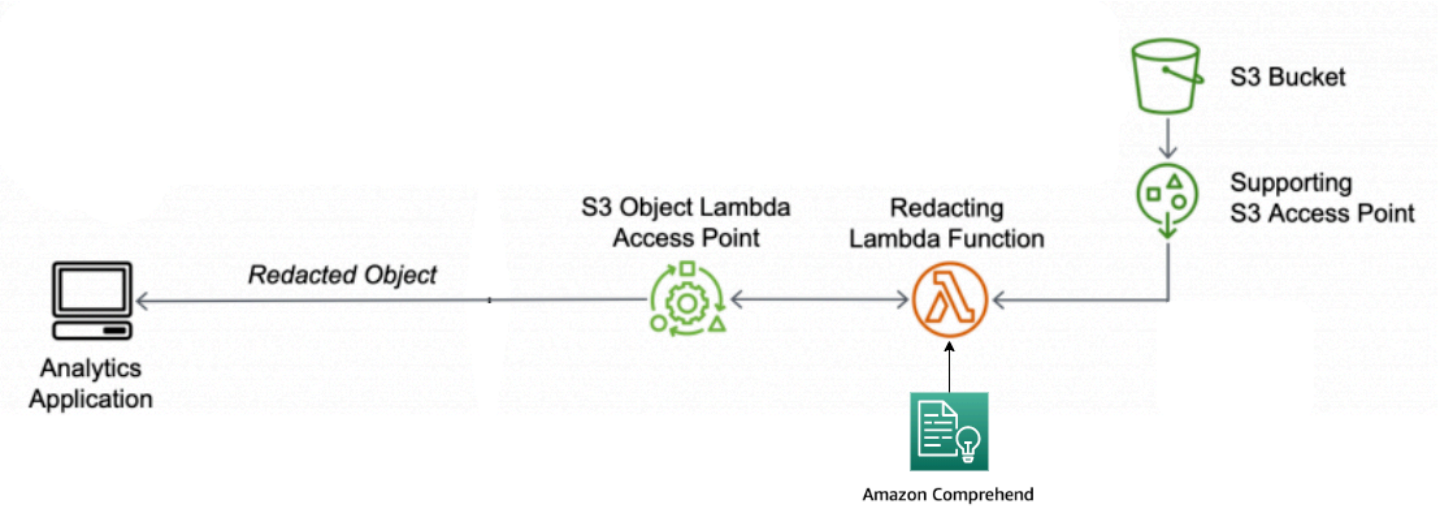
  Para obtener más información, consulte Tutorial: Uso de S3 Object Lambda para agregar marcas de agua dinámicas a las imágenes a medida que se recuperan.
- Implementar reglas de autorización personalizadas para acceder a los datos.

Para obtener más información acerca de S3 Object Lambda, consulte Transformación de objetos con Lambda para objetos S3.

Tutorial: detección y redacción de datos de PII con S3 Object Lambda y Amazon Comprehend

Cuando utiliza Amazon S3 para conjuntos de datos compartidos para varias aplicaciones y usuarios a los que acceder, es importante restringir la información privilegiada, como la información de identificación personal (PII), solo a entidades autorizadas. Por ejemplo, cuando una aplicación de marketing utiliza algunos datos que contienen PII, es posible que deba enmascarar primero los datos de PII para cumplir con los requisitos de privacidad de datos. Además, cuando una aplicación de análisis utiliza un conjunto de datos de inventario de pedidos de producción, es posible que deba redactar primero la información de la tarjeta de crédito del cliente para evitar fugas de datos no intencionadas.

Con S3 Object Lambda y una función AWS Lambda con la tecnología de Amazon Comprehend, puede proteger los datos de PII recuperados de S3 antes de devolverlos a una aplicación. En concreto, puede utilizar la función de Lambda prediseñada como función de redacción y adjuntarla a un punto de acceso de S3 Object Lambda. Cuando una aplicación (por ejemplo, una aplicación de análisis) envía solicitudes estándar S3 GET, estas solicitudes realizadas a través del punto de acceso de S3 Object Lambda invocan la función de Lambda de redacción prediseñada para detectar y redactar datos de PII recuperados de un bucket de S3 a través de un punto de acceso S3 compatible. A continuación, el punto de acceso de S3 Object Lambda devuelve el resultado redactado a la aplicación.



En el proceso, la función de Lambda prediseñada utiliza Amazon Comprehend, un servicio de procesamiento de lenguaje natural (NLP), para capturar variaciones en la forma en que se representa la PII, independientemente de cómo exista la PII en el texto (como numéricamente o como una combinación de palabras y números). Amazon Comprehend puede incluso utilizar el contexto en el texto para entender si un número de 4 dígitos es un PIN, los cuatro últimos números de un número de Seguro Social (SSN) o un año. Amazon Comprehend procesa cualquier archivo de texto en formato UTF-8 y puede proteger la información personal a escala sin afectar a la precisión. Para obtener más información, consulte ¿Qué es Amazon Comprehend? en la Guía para desarrolladores de Amazon Comprehend.

Objetivo

En este tutorial, aprenderá a utilizar S3 Object Lambda con la función de Lambda preconstruida`ComprehendPiiRedactionS3ObjectLambda`. Esta función utiliza Amazon Comprehend para detectar entidades de PII. A continuación, redacta estas entidades reemplazándolas con asteriscos. Al redactar la información personal, oculta los datos confidenciales, lo que puede ayudar con la seguridad y el cumplimiento normativo.

También aprenderá a usar y configurar una AWS LambdaFunction en la AWS Serverless Application Repository para trabajar junto con S3 Object Lambda para facilitar la implementación.

Temas

- Requisitos previos: cree un usuario de IAM con permisos
- Paso 1: Crear un bucket de S3
- Paso 2: Cargar un archivo a S3 bucket

Requisitos previos: cree un usuario de IAM con permisos

Antes de empezar este tutorial, debe tener una cuenta de AWS en la que puede iniciar sesión como usuario de AWS Identity and Access Management (usuario de IAM) con los permisos correctos.

Puede crear un usuario de IAM para el tutorial. Para completar este tutorial, el usuario de IAM debe adjuntar las siguientes políticas de IAM para acceder a los recursos de AWS y realizar acciones específicas.

> **Note**
>
> Para simplificar, este tutorial crea y utiliza un usuario de IAM. Después de completar este tutorial, recuerde Eliminación del rol de IAM. Para uso en producción, le recomendamos que siga las Prácticas recomendadas de seguridad de IAM en la Guía del usuario de IAM. Una práctica recomendada exige que los usuarios humanos utilicen la federación con un proveedor de identidades para acceder a AWS con credenciales temporales. Otra práctica recomendada es exigir a las cargas de trabajo que utilicen credenciales temporales con roles de IAM para acceder a AWS. Para obtener información sobre el uso de AWS IAM Identity Center para crear usuarios con credenciales temporales, consulte Getting started (Introducción) en la Guía del usuario de AWS IAM Identity Center.
> Este tutorial también utiliza políticas de acceso completo. Para uso en producción, le recomendamos que otorgue solo los permisos mínimos necesarios para su caso de uso, de acuerdo con las prácticas recomendadas de seguridad.

Su usuario de IAM requiere las siguientes políticas administradas de AWS:

- AmazonS3FullAccess: otorga permisos a todas las acciones de Amazon S3, incluidos los permisos para crear y utilizar un punto de acceso de Object Lambda.

- AWSLambda_FullAccess— Otorga permisos a todas las acciones de Lambda.
- AWSCloudFormationFullAccess— Otorga permisos a todos losAWS CloudFormationacciones.
- IAMFullAccess— Otorga permisos a todas las acciones de IAM.
- IAMAccessAnalyzerReadOnlyAccess— Otorga permisos para leer toda la información de acceso proporcionada por IAM Access Analyzer.

Puede adjuntar directamente estas directivas existentes al crear un usuario de IAM. Para obtener más información acerca de cómo crear un usuario de IAM, consulte Creación del primer grupo de usuarios y administradores de IAM en la guía del usuario de IAM.

Además, su usuario de IAM requiere una política administrada por el cliente. Para conceder permisos de usuario de IAM a todos los Recursos y acciones de AWS Serverless Application Repository, debe crear una política de IAM y adjuntar la política al usuario de IAM.

Crear y asociar una política a un usuario de IAM

1. Inicie sesión en AWS Management Console Management Console y abra la consola IAM en https://console.aws.amazon.com/iam/.
2. En el panel de navegación izquierdo, elija Policies (Políticas).
3. Elija Create Policy (Crear política).
4. En la pestaña Visual editor (Editor visual), para Service (Servicio), seleccione Choose a service (Elegir un servicio). Luego, elija Serverless Application Repository.
5. Para Actions (Acciones), en Manual actions) (Acciones manuales), seleccione All Serverless Application Repository actions (serverlessrepo:*) (Todas las acciones Serverless Application Repository servidor [serverlessrepo: *]) para este tutorial.

   Como práctica recomendada de seguridad, debe permitir solo aquellas acciones y recursos a los que un usuario necesita acceso. Para obtener más información, consulte la sección Prácticas recomendadas de IAM en la Guía del usuario de &IAM;.
6. Para Resources (Recursos), elija All resources (Todos los recursos) en este tutorial.

   Como práctica recomendada, debe definir permisos para recursos específicos de cuentas específicas. Si lo prefiere, puede conceder el menor privilegio mediante claves de condición. Para obtener más información, consulte Conceder privilegios mínimos en la Guía del usuario de IAM.
7. Elija Siguiente: Etiquetas.

8. Elija Siguiente: Revisar.
9. En la página Review Policy (Revisar política), especifique un nombre en el campo Name (Nombre) (por ejemplo: **`tutorial-serverless-application-repository`**) y una Description (descripción) (opcional) para la política que está creando. Revise el resumen de política para asegurarse de que ha concedido los permisos deseados y, a continuación, elija Create policy (Crear política) para guardar su nueva política.
10. En el panel de navegación izquierdo, elija Users (Usuarios). A continuación, elija el usuario de IAM para este tutorial.
11. En la página Summary (Resumen) del usuario elegido, elija la opción Permissions (Permisos) y luego Add permissions (Agregar permisos).
12. En Grant permissions (Conceder permisos), elija Attach existing policies directly (Asociar las políticas existentes directamente).
13. Seleccione la casilla de verificación situada junto a la política que acaba de crear (por ejemplo: **`tutorial-serverless-application-repository`**) y luego elija Next: Review (Siguiente: revisar).
14. En Permissions Summary (Resumen de permisos), revise el resumen de la política para asegurarse de que ha adjuntado la política deseada. A continuación, elija Add permissions (Agregar permisos).

Paso 1: Crear un bucket de S3

Cree un bucket para almacenar los datos originales que tiene previsto transformar.

Creación de un bucket

1. Inicie sesión AWS Management Console Management Console y abra la consola de Amazon S3 en https://console.aws.amazon.com/s3/.
2. En el panel de navegación izquierdo, elija Instancias.
3. Elija Crear bucket.

   Se abrirá la página Crear bucket.
4. En Nombre de bucket, escriba el nombre del bucket (por ejemplo: **`tutorial-bucket`**).

   Para obtener más información acerca de las reglas de nomenclatura del bucket de Amazon S3, consulte Reglas de nomenclatura de buckets.

5. En Región, elija la Región de AWS en la que desea que se encuentre el bucket.

   Para obtener más información acerca de bucket Region, consulte Descripción general de los buckets.

6. Para Configuración de Block Public Access para este bucket, conserve la configuración predeterminada (Bloquear todo acceso público está habilitado).

   Le recomendamos que deje todas las configuraciones habilitadas a menos que sepa que necesita desactivar una o varias de ellas para su caso de uso, como alojar un sitio web público. Para obtener más información acerca del bloqueo del acceso público, consulte Bloquear el acceso público a su almacenamiento de Amazon S3.

7. Mantenga la configuración restante establecida en los valores predeterminados.

   (Opcional) Si desea configurar opciones de bucket adicionales para el caso de uso específico, consulteCrear un bucket.

8. Elija Crear bucket.

Paso 2: Cargar un archivo a S3 bucket

Cargue un archivo de texto que contenga datos de PII conocidos de varios tipos, como nombres, información bancaria, números de teléfono y SSN, al bucket de S3 como datos originales de los que redactará PII más adelante en este tutorial.

Por ejemplo, puede cargarlo siguiendo el archivo `tutorial.txt`. Este es un ejemplo de un archivo de entrada de ejemplo de 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.
```

Para cargar un archivo en un bucket

1. Inicie sesión AWS Management Console Management Console y abra la consola de Amazon S3 en https://console.aws.amazon.com/s3/.
2. En el panel de navegación izquierdo, elija Instancias.
3. En la lista Buckets, elija el nombre del bucket que creó en elPaso 1 (por ejemplo: **`tutorial-bucket`**) para cargar el archivo.
4. En la pestaña Objetos del bucket, elija Cargar.
5. En la página Cargar, en Archivos y carpetas, elija Añadir archivos.
6. Seleccione un archivo que cargar y luego seleccione Abrir. Por ejemplo, puede cargar el archivo de ejemplo `tutorial.txt` mencionado anteriormente.
7. Seleccione Cargar.

Paso 3: Crear un punto de acceso de S3

Para utilizar un punto de acceso de S3 Object Lambda para acceder y transformar los datos originales, debe crear un punto de acceso de S3 y asociarlo con el bucket de S3 que creó en el Paso 1. El punto de acceso debe estar en la misma Región de AWS que los objetos que desea transformar.

Más adelante en este tutorial, utilizará este punto de acceso como punto de acceso de soporte para su punto de acceso de Object Lambda.

Para crear un punto de acceso

1. Inicie sesión AWS Management Console Management Console y abra la consola de Amazon S3 en https://console.aws.amazon.com/s3/.
2. En el panel de navegación, elija Puntos de acceso.
3. En la página de Puntos de acceso, elija Crear punto de acceso.
4. Introduzca el nombre que desee (por ejemplo: **`tutorial-pii-access-point`**) para el punto de acceso en el campo Nombre del punto de acceso.

   Para obtener más información sobre cómo asignar nombres a los puntos de acceso, consulte Reglas para asignar nombres a los puntos de acceso de Amazon S3.
5. En el campo Nombre del Bucket escriba el nombre del bucket que creó en el Paso 1 (por ejemplo: **`tutorial-bucket`**). A continuación, asocie dicha política al bucket de S3.

(Opcional) Puede elegir Examinar S3) para explorar su cuenta y buscar buckets. Si elige Examinar S3, seleccione el bucket que le interese y seleccione Elegir ruta para rellenar el campo Nombre del bucket con el nombre del bucket en cuestión.

6. Para Origen de la red, elija Internet.

   Para obtener más información acerca de los orígenes de red para los puntos de acceso, consulte Crear puntos de acceso restringidos a una nube privada virtual.

7. Todas las configuraciones de bloqueo de acceso público están habilitadas de forma predeterminada para los puntos de acceso. Le recomendamos que deje habilitada Block all public access (Bloquear todo el acceso público). Para obtener más información, consulte Administrar el acceso público a los puntos de acceso.

8. En el resto de configuraciones de punto de acceso, mantenga la configuración predeterminada.

   (Opcional) Puede modificar la configuración del punto de acceso para admitir el caso de uso. En este tutorial, le recomendamos que mantenga la configuración predeterminada.

   (Opcional) Si necesita administrar el acceso al punto de acceso, puede especificar una directiva de punto de acceso. Para obtener más información, consulte Ejemplos de políticas de puntos de acceso.

9. Elija Crear punto de acceso.

Paso 4: Configurar e implementar una función de Lambda prefabricada

Para redactar los datos de PII, configure e implemente la función prediseñada AWS Lambda `ComprehendPiiRedactionS3ObjectLambda` para utilizarla con el punto de acceso de S3 Object Lambda.

Para configurar e implementar la función de Lambda

1. Inicie sesión en la AWS Management Console y vea la función `ComprehendPiiRedactionS3ObjectLambda` en laAWS Serverless Application Repository.
2. Para Application settings (Configuración de la aplicación), en Application name (Nombre de la aplicación), conserve el valor predeterminado (`ComprehendPiiRedactionS3ObjectLambda`) para este tutorial.

(Opcional) Puede introducir el nombre que desea dar a esta aplicación. Es posible que desee hacerlo si tiene previsto configurar varias funciones de Lambda para diferentes necesidades de acceso para el mismo conjunto de datos compartido.

3. Para MaskCharacter, conserve el valor predeterminado (*****). El carácter de máscara reemplaza cada carácter de la entidad PII redactada.
4. Para MaskMode, conserve el valor predeterminado (MASK). El valor MaskMode especifica si la entidad PII se redacta con el carácter `MASK` o el valor `PII_ENTITY_TYPE`.
5. Para redactar los tipos de datos especificados, paraPiiEntityTypes, conserve el valor predeterminadoALL. El valor PiiEntityTypes especifica los tipos de entidad de PII que se deben considerar para la redacción.

   Para obtener más información acerca de la lista de tipos de entidad de PII admitidos, consulte Detecte información de identificación personal (PII) en la Guía para desarrolladores de Amazon Comprehend.
6. Mantenga la configuración restante establecida en los valores predeterminados.

   (Opcional) Si desea configurar opciones adicionales para el caso de uso específico, consulte la sección Readme file (Archivo Léame) en el lado izquierdo de la página.
7. Elija la casilla de verificación situada junto a I acknowledge that this app creates custom IAM roles (Confirmo que esta aplicación puede crear roles de IAM personalizados).
8. Elija Implementar.
9. En la página de la nueva aplicación, en Resources (Recursos), elija la opción Logical ID (ID lógico) de la función de Lambda que implementó para revisar la función en la página de funciones de Lambda.

Paso 5: Crear un punto de acceso de S3 Object Lambda

Un punto de acceso S3 Object Lambda proporciona la flexibilidad para invocar una función de Lambda directamente desde una solicitud GET S3 para que la función pueda redactar datos de PII recuperados de un punto de acceso de S3. Al crear y configurar un punto de acceso de S3 Object Lambda, debe especificar la función de Lambda redactante para invocar y proporcionar el contexto de evento en formato JSON como parámetros personalizados para que los utilice Lambda.

S3 Object Lambda proporciona contexto sobre la solicitud que se realiza en el evento pasado a Lambda. Para obtener más información sobre los campos en el contexto de evento, consulte Formato y uso del contexto del evento.

Para crear un punto de acceso de S3 Object Lambda

1. Inicie sesión AWS Management Console Management Console y abra la consola de Amazon S3 en https://console.aws.amazon.com/s3/.
2. En el panel de navegación del lado izquierdo de la consola, elija Object Lambda Access Points (Puntos de acceso de Object Lambda).
3. En la página Object Lambda Access Points (Puntos de acceso Object Lambda), elija Create Object Lambda Access Point (Crear un punto de acceso de Object Lambda).
4. En Nombre del punto de acceso del objeto Lambda, introduzca el nombre que desea utilizar para el punto de acceso del objeto Lambda (por ejemplo: **`tutorial-pii-object-lambda-accesspoint`**).
5. Para Supporting Access Point (Soporte de punto de acceso), introduzca o busque el punto de acceso estándar que ha creado en el Paso 3(por ejemplo: **`tutorial-pii-access-point`**), y luego elija Choose Supporting Access Point (Elegir soporte de punto de acceso).
6. Para las S3 APIs (API de S3), para recuperar objetos del bucket de S3 para que la función de Lambda los procese, seleccione GetObject.
7. Para Invoke Lambda function (Invocación de una función de Lambda), puede elegir una de las dos opciones siguientes.
   - Elegir Choose from functions in your account (Elija entre las funciones de su cuenta) y elija la función de Lambda que implementó en el Paso 4 (por ejemplo: **`serverlessrepo-ComprehendPiiRedactionS3ObjectLambda`**) desde la lista desplegable Lambda function (Función de Lambda).
   - Elegir Enter ARN (Ingresar ARN) y, a continuación, introduzca el nombre de recurso de Amazon (ARN) de la función de Lambda que ha creado en el Paso 4.
8. Para Lambda function version (Versión de función de Lambda), elija $LATEST (la última versión de la función de Lambda que implementó en el Paso 4).
9. (Opcional) Si necesita su función de Lambda para reconocer y procesar solicitudes GET con encabezados de rango y número de pieza, seleccione Lambda function supports requests using range (La función de Lambda admite solicitudes usando rango) y Lambda function supports requests using part numbers (La función de Lambda admite solicitudes usando números de pieza). De lo contrario, desactive estas dos casillas de verificación.

Para obtener más información acerca de cómo utilizar números de rango o rango con S3 Object Lambda, consulte Trabajar con encabezados Range y partNumber.

10. (Opcional) En Payload - optional (Carga - opcional), agregue texto JSON para proporcionar información adicional a su función de Lambda.

    Una carga es texto JSON opcional que puede proporcionar a su función de Lambda como entrada para todas las invocaciones procedentes de un punto de acceso de S3 Object Lambda específico. Puede configurar cargas con diferentes parámetros para diferentes puntos de acceso Object Lambda que invoquen la misma función de Lambda, ampliando así la flexibilidad de su función de Lambda.

    Para obtener más información acerca de patones de rutas, consulte Formato y uso del contexto del evento.

11. (Opcional) Para las Métricas de solicitudes - opcional, elija Deshabilitar o Habilitar para agregar la supervisión de Amazon S3 al punto de acceso del objeto Lambda. Las métricas de solicitud se facturan según la tarifa de Amazon CloudWatch estándar. Para obtener más información, consulte los precios de CloudWatch.
12. En Object Lambda Access Point policy - optional (Política de punto de acceso Object Lambda - opcional), conserve la configuración predeterminada.

    (Opcional) Ejecute para establecer la política de recursos. Esta política de recursos concede el permiso de API `GetObject` para usar el punto de acceso del objeto Lambda especificado.

13. Mantenga la configuración restante establecida en los valores predeterminados y elija Create Object Lambda Access Point (Creación de punto de acceso Object Lambda).

Paso 6: Utilizar el punto de acceso de S3 Object Lambda para recuperar el archivo redactado

Ahora, S3 Object Lambda está listo para redactar los datos de PII de su archivo original.

Para utilizar el punto de acceso de S3 Object Lambda para recuperar el archivo redactado

Cuando solicita recuperar un archivo a través de su punto de acceso de S3 Object Lambda, cree una llamada de la API `GetObject` a S3 Object Lambda. S3 Object Lambda invoca la función de Lambda para redactar sus datos PII y devuelve los datos transformados como la respuesta al estándar S3`GetObject`Llamada a la API.

1. Inicie sesión AWS Management Console Management Console y abra la consola de Amazon S3 en https://console.aws.amazon.com/s3/.
2. En el panel de navegación, elija Object Lambda Access Points (Puntos de acceso de Object Lambda).
3. En la página Puntos de acceso del objeto Lambda, elija el punto de acceso de S3 Object Lambda que creó en el Paso 5 (por ejemplo, **`tutorial-pii-object-lambda-accesspoint`**).
4. En la pestaña Objetos de su punto de acceso de S3 Object Lambda, seleccione el archivo que tenga el mismo nombre (por ejemplo, `tutorial.txt`) como el que cargó en el bucket de S3 en el Paso 2.

   Este archivo debe contener todos los datos transformados.
5. Para ver los datos transformados, elija Open (Abrir) o Download (Descargar).

   Debería poder ver el archivo redactado, tal como se muestra en el siguiente ejemplo.

```
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 **********************.
```

Paso 7: Limpiar

Si redactó sus datos mediante S3 Object Lambda solo como parte de un ejercicio de aprendizaje, elimine los recursos de AWS que asignó para dejar de acumular cargos.

Pasos secundarios

- Eliminar el punto de acceso del objeto Lambda
- Elimine el punto de acceso de S3

- Para eliminar la función de Lambda
- Eliminación del grupo de registros de CloudWatch
- Elimine el archivo original en el bucket de origen de S3
- Eliminar el bucket de origen de S3
- Elimine el rol de IAM para su función de Lambda
- Eliminar la política administrada por el cliente para su usuario de IAM
- Eliminación del rol de IAM

Eliminar el punto de acceso del objeto Lambda

1. Inicie sesión AWS Management Console Management Console y abra la consola de Amazon S3 en https://console.aws.amazon.com/s3/.
2. En el panel de navegación, elija Object Lambda Access Points (Puntos de acceso de Object Lambda).
3. En la página Puntos de acceso del objeto lambda, elija el botón de opción situado a la izquierda del punto de acceso de S3 Object Lambda que creó en el Paso 5 (por ejemplo, **`tutorial-pii-object-lambda-accesspoint`**).
4. Elija Delete (Eliminar).
5. Confirme que desea eliminar el punto de acceso del objeto Lambda. Para ello, escriba su nombre en el campo de texto que aparece y elija Eliminar.

Elimine el punto de acceso de S3

1. Inicie sesión AWS Management Console Management Console y abra la consola de Amazon S3 en https://console.aws.amazon.com/s3/.
2. En el panel de navegación, elija Access Points (Puntos de acceso).
3. Desplácese hasta el punto de acceso que creó en el Paso 3 (por ejemplo: **`tutorial-pii-access-point`**) y elija el botón de opción situado junto al nombre del punto de acceso.
4. Elija Eliminar.
5. Confirme que desea eliminar el punto de acceso escribiendo su nombre en el campo de texto que aparece y elija Delete (Eliminar).

Para eliminar la función de Lambda

1. En la consola AWS Lambda en https://console.aws.amazon.com/lambda/, elija Functions (Funciones) en el panel de navegación izquierdo.
2. Elija la función que creó en el paso 4 (por ejemplo: **`serverlessrepo-ComprehendPiiRedactionS3ObjectLambda`**).
3. Elija Acciones y, a continuación, elija Eliminar.
4. En el cuadro de diálogo Eliminar función, elija Eliminar.

Eliminación del grupo de registros de CloudWatch

1. Abra la consola de CloudWatch en https://console.aws.amazon.com/cloudwatch/.
2. En el panel de navegación izquierdo, elija Logs (Registros), Log groups (Grupos de registros).
3. Busque el grupo de registros cuyo nombre termina con la función de Lambda que creó en elPaso 4(por ejemplo: **`serverlessrepo-ComprehendPiiRedactionS3ObjectLambda`**).
4. Elija Actions (Acciones) y, a continuación, elija Delete log group (Eliminar grupo de registro).
5. En el cuadro de diálogo Delete log group(s), Eliminar grupo(s) de registro(s) elija Delete (Eliminar).

Elimine el archivo original en el bucket de origen de S3

1. Inicie sesión AWS Management Console Management Console y abra la consola de Amazon S3 en https://console.aws.amazon.com/s3/.
2. En el panel de navegación izquierdo, elija Instancias.
3. En la lista Bucket name (Nombre del Bucket), seleccione el nombre del bucket al que ha subido el archivo original en el Paso 2 (por ejemplo: **`tutorial-bucket`**).
4. Seleccione la casilla de verificación situada a la izquierda de los nombres de los objetos que desea eliminar (por ejemplo: `tutorial.txt`).
5. Elija Eliminar.
6. En la página Delete objects (Eliminar objetos), en la sección Permanently delete objects? (¿Eliminar objetos de forma permanente?), confirme que desea eliminar este objeto escribiendo **`permanently delete`** en el cuadro de texto.
7. Elija Eliminar objetos.

Eliminar el bucket de origen de S3

1. Inicie sesión AWS Management Console Management Console y abra la consola de Amazon S3 en https://console.aws.amazon.com/s3/.
2. En el panel de navegación izquierdo, elija Instancias.
3. En la lista Buckets, elija el botón de opción situado junto al nombre del bucket que creó en el Paso 1 (por ejemplo: **`tutorial-bucket`**).
4. Elija Eliminar.
5. En la página Delete bucket (Eliminar bucket) confirme que desea eliminar el bucket introduciendo el nombre del bucket en el campo de texto y, a continuación, elija Delete bucket (Eliminar bucket).

Elimine el rol de IAM para su función de Lambda

1. Inicie sesión en AWS Management Console Management Console y abra la consola IAM en https://console.aws.amazon.com/iam/.
2. En el panel de navegación, elija Roles y, a continuación, seleccione la casilla de verificación junto al nombre del rol que desee eliminar. El nombre del rol comienza con el nombre de la función de Lambda que ha implementado en el Paso 4 (por ejemplo: **`serverlessrepo-ComprehendPiiRedactionS3ObjectLambda`**).
3. Elija Eliminar.
4. En el cuadro de diálogo Delete (Eliminar), escriba el nombre del rol en el campo de entrada de texto para confirmar la eliminación. A continuación, elija Delete (Eliminar).

Eliminar la política administrada por el cliente para su usuario de IAM

1. Inicie sesión en AWS Management Console Management Console y abra la consola IAM en https://console.aws.amazon.com/iam/.
2. En el panel de navegación izquierdo, elija Policies (Políticas).
3. En la página Policies (Políticas), introduzca el nombre de la política administrada por el cliente que ha creado en Prerequisites (Requisitos previos)(por ejemplo: **`tutorial-serverless-application-repository`**) en el cuadro de búsqueda para filtrar la lista de políticas. Seleccione el botón de opción situado junto al nombre del punto de acceso que desea eliminar.
4. Elija Actions (Acciones) y, a continuación, elija Delete (Eliminar).

5. Confirme que desea eliminar el punto de acceso escribiendo su nombre en el campo de texto que aparece y elija Delete (Eliminar).

Eliminación del rol de IAM

1. Inicie sesión en AWS Management Console Management Console y abra la consola IAM en https://console.aws.amazon.com/iam/.
2. En el panel de navegación, elija Roles y, a continuación, seleccione la casilla de verificación junto al nombre del rol que desee eliminar.
3. En la parte superior de la página, elija Delete user (Eliminar usuario).
4. En el cuadro de diálogo Delete ***`user name`***? (¿Eliminar Nombre de usuario?), introduzca el nombre de usuario en el campo de entrada de texto para confirmar la eliminación del usuario. Elija Eliminar.

Siguientes pasos

Después de completar este tutorial, puede explorar más a fondo los siguientes casos de uso relacionados:

- Puede crear varios puntos de acceso de S3 Object Lambda y habilitarlos con funciones de Lambda prediseñadas que se configuran de manera diferente para redactar tipos específicos de PII en función de las necesidades empresariales de los mecanismos de acceso a los datos.

  Cada tipo de usuario asume un rol de IAM y solo tiene acceso a un punto de acceso de S3 Object Lambda (administrado a través de políticas de IAM). A continuación, adjuntar cada función de Lambda `ComprehendPiiRedactionS3ObjectLambda` configurada para un caso de uso de redacción diferente a un punto de acceso de S3 Object Lambda. Para cada punto de acceso de S3 Object Lambda, puede tener un punto de acceso S3 compatible para leer datos de un bucket de S3 que almacene el conjunto de datos compartido.

  Para obtener más información acerca de cómo crear una directiva de bucket de S3 que permita a los usuarios leer desde el bucket solo a través de puntos de acceso de S3, consulte Configurar las políticas de IAM para el uso de puntos de acceso.

  Para obtener más información acerca de cómo conceder a un usuario permiso para acceder a la función de Lambda, el punto de acceso de S3 y el punto de acceso de S3 Object Lambda, consulte Configuración de las políticas de IAM para puntos de acceso de Object Lambda.

- Puede construir su propia función de Lambda y usar S3 Object Lambda con su función de Lambda personalizada para satisfacer sus necesidades de datos específicas.

  Por ejemplo, para explorar varios valores de datos, puede usar S3 Object Lambda y su propia función de Lambda que utiliza características de Amazon Comprehend, como el reconocimiento de entidades, el reconocimiento de frases clave, el análisis de sentimientos y la clasificación de documentos, para procesar datos. También puede utilizar S3 Object Lambda junto conAmazon Comprehend Medical, un servicio PNL apto para HIPAA, para analizar y extraer datos de una manera consciente del contexto.

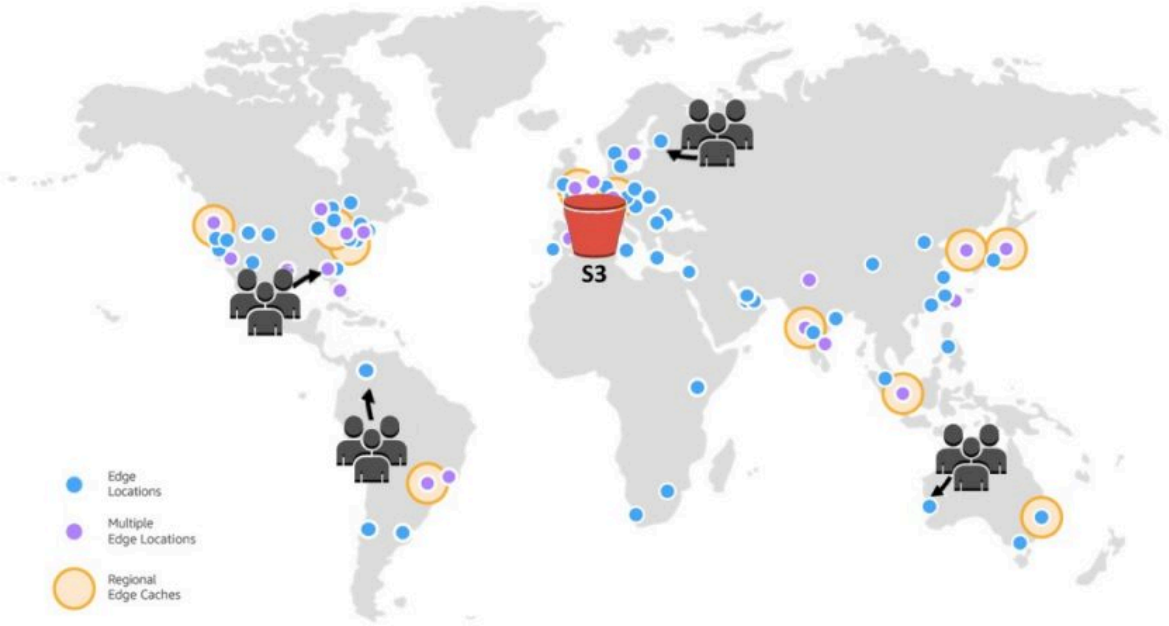
  Para obtener más información acerca de cómo transformar datos con S3 Object Lambda y su propia función de Lambda, consulteTutorial: transformación de datos para su aplicación con S3 Object Lambda.

Tutorial: Alojamiento de video en streaming bajo demanda con Amazon S3, Amazon CloudFront y Amazon Route 53

Puede utilizar Amazon S3 con Amazon CloudFront para alojar videos y visualizarlos bajo demanda con seguridad y escalabilidad. Streaming de video bajo demanda (VOD) significa que el contenido de video se almacena en un servidor, y los usuarios pueden verlo en cualquier momento.

CloudFront es un servicio de red de entrega de contenido (CDN) rápido, altamente seguro y programable. CloudFront puede entregar su contenido de forma segura a través de HTTPS desde todas las ubicaciones de borde de CloudFront en el mundo. Para obtener más información acerca de CloudFront, consulte ¿Qué es Amazon CloudFront? en la Guía para desarrolladores de Amazon CloudFront.

El almacenamiento en caché de CloudFront reduce la cantidad de solicitudes a las que debe responder directamente el servidor de origen. Cuando un espectador (usuario final) solicita un video servido por usted con CloudFront, la solicitud se dirige a una ubicación periférica próxima más cercana a donde se encuentra el espectador. CloudFront sirve el video desde su caché y lo recupera del bucket de S3 solo si aún no está almacenado en caché. Esta característica de administración acelera la entrega de video a los usuarios de todo el mundo con baja latencia, alto rendimiento y altas velocidades de transferencia. Para obtener más información acerca de la administración del almacenamiento en caché de CloudFront, consulte Optimización del almacenamiento en caché y la disponibilidad en la Guía para desarrolladores de Amazon CloudFront.



Objetivo

En este tutorial, configura un bucket de S3 para alojar el streaming de video bajo demanda mediante CloudFront para la entrega y Amazon Route 53 para el sistema de nombres de dominio (DNS) y la administración de dominios personalizados.

Temas

Requisitos previos: registrar y configurar un dominio personalizado con Route 53

Antes de comenzar este tutorial, debe registrar y configurar un dominio personalizado (por ejemplo, **`example.com`**) con Route 53 para configurar su distribución de CloudFront para que use un nombre de dominio personalizado más adelante.

Sin un nombre de dominio personalizado, el video de S3 es accesible públicamente y está alojado a través de CloudFront en una URL similar a la siguiente:

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

Por ejemplo, **`https://d111111abcdef8.cloudfront.net/sample.mp4`**.

Después de configurar la distribución de CloudFront para utilizar un nombre de dominio personalizado configurado con Route 53, el video de S3 será accesible públicamente y estará alojado a través de CloudFront en una dirección URL similar a la siguiente:

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

Por ejemplo, **`https://www.example.com/sample.mp4`**. A los espectadores les resultará más sencillo e intuitivo usar un nombre de dominio personalizado.

Para registrar el nombre de un dominio, consulte Registro de nombres de dominio mediante Route 53 en la Guía para desarrolladores de Amazon Route 53.

Cuando registra un nombre de dominio con Route 53, Route 53 crea la zona alojada para usted, que usará más adelante en este tutorial. Esta zona alojada es el lugar donde almacena la información acerca de cómo dirigir el tráfico de su dominio; por ejemplo, a una instancia de Amazon EC2 o una distribución de CloudFront.

Hay tarifas asociadas al registro de dominios, la zona alojada y las consultas de DNS que recibe su dominio. Para obtener más información, consulte Precios de Amazon Route 53.

> **Note**
>
> Cuando registra un dominio, cuesta dinero inmediatamente y es irreversible. Puede elegir no renovar automáticamente el dominio, pero paga por adelantado y lo posee durante el año. Para obtener más información, consulte Renovación de un nuevo dominio en la Guía para desarrolladores de Amazon Route 53.

Paso 1: crear un bucket de S3

Debe crear un bucket para almacenar el video original que planea transmitir.

Creación de un bucket

1. Inicie sesión AWS Management Console Management Console y abra la consola de Amazon S3 en https://console.aws.amazon.com/s3/.
2. En el panel de navegación izquierdo, elija Instancias.
3. Elija Crear bucket.

   Se abrirá la página Create bucket (Crear bucket).
4. En Bucket name (Nombre de bucket), ingrese el nombre del bucket (por ejemplo, **`tutorial-bucket`**).

   Para obtener más información acerca de las reglas de nomenclatura del bucket de Amazon S3, consulte Reglas de nomenclatura de buckets.
5. En Region (Región), elija la Región de AWS en la que desea que se encuentre el bucket.

   Si es posible, debe elegir la ubicación de la región que probablemente esté más cerca de la mayoría de sus usuarios. Para obtener más información acerca de bucket Region, consulte Descripción general de los buckets.
6. Para Configuración de Block Public Access para este bucket, conserve la configuración predeterminada (Bloquear todo acceso público está habilitado).

   Incluso con la opción Block all public access (Bloquear todo el acceso público) habilitada, los usuarios pueden acceder al video cargado a través de CloudFront. Esta característica es una ventaja importante de utilizar CloudFront para alojar un video almacenado en S3.

Le recomendamos que deje todas las configuraciones habilitadas, a menos que sepa que necesita desactivar una o varias de ellas para su caso de uso. Para obtener más información acerca del bloqueo del acceso público, consulte Bloquear el acceso público a su almacenamiento de Amazon S3.

7. Mantenga la configuración restante establecida en los valores predeterminados.

   (Opcional) Si desea configurar opciones de bucket adicionales para el caso de uso específico, consulteCrear un bucket.

8. Elija Crear bucket.

Paso 2: Cargar un video en el bucket de S3

En el siguiente procedimiento, se describe cómo cargar un archivo de video a un bucket de S3 mediante la consola. Cuando carga muchos videos grandes a S3, también puede usar Amazon S3 Transfer Acceleration para configurar transferencias de archivos rápidas y seguras. Transfer Acceleration puede acelerar la carga de video en su bucket de S3 para transferir a larga distancia videos más grandes. Para obtener más información, consulte Configuración de transferencias de archivos rápidas y seguras con Amazon S3 Transfer Acceleration.

Para cargar un archivo en el bucket

1. Inicie sesión AWS Management Console Management Console y abra la consola de Amazon S3 en https://console.aws.amazon.com/s3/.
2. En el panel de navegación izquierdo, elija Instancias.
3. En la lista Buckets, elija el nombre del bucket que creó en elPaso 1 (por ejemplo: **`tutorial-bucket`**) para cargar el archivo.
4. En la pestaña Objetos del bucket, elija Cargar.
5. En la página Cargar, en Archivos y carpetas, elija Añadir archivos.
6. Seleccione un archivo que cargar y luego seleccione Abrir.

   Por ejemplo, puede cargar un archivo de video denominado `sample.mp4`.

7. Seleccione Upload (Cargar).

Paso 3: Cree una identidad de acceso de origen de CloudFront

Para restringir el acceso directo al video desde el bucket de S3, debe crear un usuario de CloudFront especial denominado identidad de acceso de origen (OAI). Asociará la OAI a su distribución más adelante en este tutorial. Al usar una OAI, se asegura de que los usuarios no puedan omitir CloudFront y obtengan el video directamente desde el bucket de S3. Solo la OAI de CloudFront puede acceder al archivo en el bucket de S3. Para obtener más información, consulte Restricción del acceso a contenido de Amazon S3 mediante una OAI en la Guía para desarrolladores de Amazon CloudFront.

Para crear una OAI de CloudFront

1. Inicie sesión en la AWS Management Console y abra la consola de CloudFront en https://console.aws.amazon.com/cloudfront/v4/home.
2. En el panel de navegación izquierdo, en la sección Seguridad, elija Acceso de origen.
3. En la pestaña Identidades, elija Crear identidad de acceso de origen.
4. Ingrese un nombre (por ejemplo, **`S3-OAI`**) para la nueva identidad de acceso de origen.
5. Seleccione Create (Crear).

Paso 4: crear una distribución de CloudFront

Para utilizar CloudFront para servir y distribuir el video en su bucket de S3, debe crear una distribución de CloudFront.

Pasos secundarios

- Crear una distribución de CloudFront.
- Revisar la política de bucket

Crear una distribución de CloudFront.

1. Inicie sesión en la AWS Management Console y abra la consola de CloudFront en https://console.aws.amazon.com/cloudfront/v4/home.
2. En el panel de navegación izquierdo, elija Distributions (Distribuciones).
3. Elija Create distribution (Crear distribución).

4. En la sección Origin (Origen), para Origin domain (Dominio de origen), elija el nombre de dominio de su origen de S3, que comienza con el nombre del bucket de S3 que creó en el Paso 1 (por ejemplo: **`tutorial-bucket`**).
5. Para Acceder al origen, seleccione Identidades de acceso antiguas.
6. En Origin access identity (Identidad de acceso de origen), elija la identidad de acceso de origen que creó en el Paso 3 (por ejemplo, **`S3-OAI`**).
7. En Bucket policy (Política de bucket), elija Yes, update the bucket policy (Sí, actualizar la política de bucket).
8. En la sección Default cache behavior (Comportamiento de caché predeterminado), en la política de protocolo del usuario, elija Redirect HTTP to HTTPS (Redireccionamiento de HTTP a HTTPS).

   Cuando elige esta característica, las solicitudes HTTP se redirigen automáticamente a HTTPS para proteger su sitio web y los datos de sus usuarios.
9. Para las otras opciones de configuración de la sección Default Cache Behavior Settings (Configuración del comportamiento de caché predeterminado), mantenga los valores predeterminados.

   (Opcional) Puede controlar el tiempo en el que se mantienen los archivos en una caché de CloudFront antes de que CloudFront reenvíe otra solicitud al origen. Reducir la duración le permite ofrecer contenido dinámico. Aumentar la duración implica que sus usuarios podrán disfrutar de un mejor rendimiento, ya que es más probable que los archivos se ofrezcan directamente desde la caché de borde. Una mayor duración también reduce la carga en el origen. Para obtener más información, consulte Administración de cuánto tiempo se mantiene el contenido en una caché perimetral (vencimiento) en la Guía para desarrolladores de Amazon CloudFront.
10. En las otras secciones, conserve la configuración restante establecida en los valores predeterminados.

    Para obtener más información sobre estas opciones de configuración, consulte Valores que deben especificarse al crear o actualizar una distribución en la Guía para desarrolladores de Amazon CloudFront.
11. En la parte inferior de la página, elija Create Distribution (Crear distribución).
12. En la pestaña General para la distribución de CloudFront, en Details (Detalles), el valor de la columna Last modified (Última modificación) para la distribución cambia de Deploying

(Implementación) a la marca temporal en que se modificó la distribución por última vez. Este proceso normalmente dura unos minutos.

Revisar la política de bucket

1. Inicie sesión AWS Management Console Management Console y abra la consola de Amazon S3 en https://console.aws.amazon.com/s3/.
2. En el panel de navegación izquierdo, elija Instancias.
3. En la lista Buckets, elija el nombre del bucket que utilizó anteriormente como origen de la distribución de CloudFront (por ejemplo: **`tutorial-bucket`**).
4. Elija la pestaña Permissions (Permisos).
5. En la sección Bucket policy (Política de bucket), confirme que aparece una instrucción similar a la siguiente en el texto de la política de bucket:

```
{
    "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/*"
        }
    ]
}
```

Esta es la declaración que su distribución de CloudFront agrega a su política de bucket cuando elige Yes, update the bucket policy (Sí, actualizar la política de bucket).

Esta actualización de la política de bucket indica que configuró correctamente la distribución de CloudFront para restringir el acceso al bucket de S3. Debido a esta restricción, solo se puede acceder a los objetos del bucket a través de su distribución de CloudFront.

Paso 5: Acceda al video a través de la distribución de CloudFront

Ahora, CloudFront puede servir el video almacenado en el bucket de S3. Para acceder a su video a través de CloudFront, debe combinar el nombre de dominio de distribución de CloudFront con la ruta de acceso al video en el bucket de S3.

Para crear una dirección URL para el video de S3 con el nombre de dominio de distribución de CloudFront

1. Inicie sesión en la AWS Management Console y abra la consola de CloudFront en https://console.aws.amazon.com/cloudfront/v4/home.
2. En el panel de navegación izquierdo, elija Distributions (Distribuciones).
3. Para obtener el nombre de dominio de distribución, haga lo siguiente:
   a. En la columna Origins (Orígenes), identifique la distribución de CloudFront; para ello, busque el nombre de origen, que comienza con el bucket de S3 que creó en el Paso 1 (por ejemplo, **`tutorial-bucket`**).
   b. Después de encontrar la distribución de la lista, amplíe la columna Domain name (Nombre del dominio) para copiar el valor del nombre del dominio para la distribución de CloudFront.
4. En una nueva pestaña del navegador, pegue el nombre del dominio de distribución que copió anteriormente.
5. Vuelva a la pestaña anterior del navegador y abra la consola de S3 en https://console.aws.amazon.com/s3/.
6. En el panel de navegación situado a la izquierda, elija Buckets.
7. En la lista Buckets, seleccione el nombre del bucket que creó en el paso 1 (por ejemplo: **`tutorial-bucket`**).
8. En la lista Objects (Objetos), elija el nombre del video que cargó en el Paso 2 (por ejemplo, `sample.mp4`).
9. En la página de detalles del objeto, sección Object overview (Información general del objeto), copie el valor de la clave. Este valor es la ruta al objeto de video que se cargó en el bucket de S3.
10. Vuelva a la pestaña del navegador en la que pegó previamente el nombre de dominio de distribución, ingrese una barra diagonal (**`/`**) después del nombre de dominio de distribución y pegue la ruta al video que copió anteriormente (por ejemplo, `sample.mp4`).

Ahora, su video de S3 es accesible públicamente y está alojado a través de CloudFront en una URL similar a la siguiente:

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

Reemplace el *nombre de dominio de distribución de CloudFront* y la *ruta al video de S3* con los valores apropiados. La dirección URL de ejemplo es: **https://d111111abcdef8.cloudfront.net/sample.mp4**

Paso 6: Configure su distribución de CloudFront para usar el nombre de dominio personalizado

Para utilizar su propio nombre de dominio en lugar del nombre de dominio de CloudFront en la URL para acceder al video de S3, agregue un nombre de dominio alternativo a la distribución de CloudFront.

Pasos secundarios

- Solicite un certificado SSL
- Agregue un nombre de dominio alternativo a su distribución de CloudFront.
- Cree un registro de DNS para enrutar el tráfico de su nombre de dominio alternativo al nombre de dominio de su distribución de CloudFront
- Verifique si IPv6 está habilitado para su distribución y cree otro registro de DNS si es necesario

Solicite un certificado SSL

Para permitir que los usuarios utilicen HTTPS y su nombre de dominio personalizado en la URL de la transmisión de video, utilice AWS Certificate Manager (ACM) para solicitar un certificado de capa de conexión segura (SSL). El certificado de SSL establece una conexión de red cifrada al sitio web.

1. Inicie sesión en AWS Management Console y abra la consola de ACM en https://console.aws.amazon.com/cloudfront/.
2. Si aparece la página de introducción, en Provision certificates (Aprovisionar certificados), elija Get Started (Comenzar).
3. En la página Request a certificate (Solicitar un certificado), elija Request a public certificate (Solicitar un certificado público) y luego, Request a certificate (Solicitar un certificado).

4. En la página Add domain names (Agregar nombres de dominio), ingrese el nombre de dominio completo del sitio (FQDN) que desea proteger con un certificado de SSL/TLS. Utilice un asterisco (*) para solicitar un certificado comodín que proteja varios nombres de sitios del mismo dominio. Para este tutorial, ingrese ***** y el nombre de dominio personalizado que configuró en Prerequisites (Requisitos previos). Por ejemplo, ingrese ***.example.com** y luego, elija Next (Siguiente).

   Para obtener más información, consulte Para solicitar un certificado público de ACM (consola) en la AWS Certificate ManagerGuía del usuario.

5. En la página Select validation method (Seleccionar método de validación), elija DNS validation (Validación por DNS). A continuación, elija Siguiente.

   Si no puede editar su configuración de DNS, recomendamos que utilice la validación de dominios de DNS en lugar de la validación por correo electrónico. La validación por DNS presenta varios beneficios con respecto a la validación por correo electrónico. Para obtener más información, consulte Opción 1: validación de DNSen la Guía del usuario de AWS Certificate Manager.

6. (Opcional) En la página Add tags (Agregar etiquetas), etiquete el certificado con metadatos.
7. Elija Revisar.
8. En la página Review (Revisar), verifique que la información en Domain name (Nombre del dominio) y Validation methods (Método de validación) sea la correcta. Elija Confirm and request.

   En la página Validation (Validación), se muestra que su solicitud se está procesando y que los dominios de certificado se están validando. Los certificados pendientes de validación se encuentran en el estado Pending validation (Pendiente de validación).

9. En la página Validation (Validación), elija la flecha hacia abajo situada a la izquierda del nombre de dominio personalizado y luego, Create record in Route 53 (Crear registro en Route 53) para validar la propiedad del dominio a través de DNS.

   Esto agrega un registro CNAME proporcionado por AWS Certificate Manager a la configuración de DNS.

10. En el cuadro de diálogo Create record in Route 53 (Crear registro en Route 53), elija Create (Crear).

    La página Validation (Validación) debería mostrar una notificación de estado Success (Correcto) en la parte inferior.

11. Seleccione Continue (Continuar) para ver la página de lista Certificates (Certificados).

El estado para su nuevo certificado cambiará de Pending validation (Pendiente de validación) a Issued (Emitido) en 30 minutos.

Agregue un nombre de dominio alternativo a su distribución de CloudFront.

1. Inicie sesión en la AWS Management Console y abra la consola de CloudFront en https://console.aws.amazon.com/cloudfront/v4/home.
2. En el panel de navegación izquierdo, elija Distributions (Distribuciones).
3. Elija el ID de la distribución que creó en el paso 4.
4. En la pestaña General, vaya a la sección Settings (Configuración) y elija Edit (Editar).
5. En la página Edit settings (Editar configuración), para Alternate domain name (CNAME) - optional (Nombre de dominio alternativo (CNAME): opcional), elija Add item (Agregar elemento) para agregar los nombres de dominio personalizados que desee usar en las direcciones URL para el video de S3 servido por esta distribución de CloudFront.

   En este tutorial, por ejemplo, si desea dirigir el tráfico de un subdominio, como `www.example.com`, ingrese el nombre del subdominio (`www`) con el nombre de dominio (`example.com`). En concreto, escriba **`www.example.com`**.

   > **Note**
   >
   > El nombre de dominio alternativo (CNAME) que agregue debe estar cubierto por el certificado SSL que previamente adjuntó a la distribución de CloudFront.

6. Para Custom SSL certificate - optional (Certificado de SSL personalizado: opcional), elija el certificado de SSL que solicitó anteriormente (por ejemplo, **`*.example.com`**).

   > **Note**
   >
   > Si no ve el certificado de SSL inmediatamente después de solicitarlo, espere 30 minutos y actualice la lista hasta que el certificado de SSL esté disponible.

7. Mantenga la configuración restante establecida en los valores predeterminados. Elija Guardar cambios.

8. En la pestaña General para la distribución, espere hasta que el valorLast modified (Última modificación) cambie de Deploying (Implementación) a la marca temporal en que se modificó la distribución por última vez.

Cree un registro de DNS para enrutar el tráfico de su nombre de dominio alternativo al nombre de dominio de su distribución de CloudFront

1. Inicie sesión en la AWS Management Console y abra la consola de Route 53 en https://console.aws.amazon.com/route53/.
2. En el panel de navegación, elija Hosted zones (Zonas alojadas).
3. En la página Hosted zones (Zonas alojadas), elija el nombre de la zona alojada que Route 53 creó por usted en Prerequisites (Requisitos previos) (por ejemplo, **`example.com`**).
4. Elija Create record (Crear registro) y luego utilice el método de registro de creación rápida.
5. Para Record name (Nombre del registro), mantenga el valor del nombre del registro igual que el nombre de dominio alternativo de la distribución de CloudFront que agregó anteriormente.

   En este tutorial, para dirigir el tráfico a un subdominio, como `www.example.com`, introduzca el nombre del subdominio sin el nombre de dominio. Por ejemplo, escriba solo **`www`** en el campo de texto anterior al nombre de dominio personalizado.
6. En Record type (Tipo de registro), elija A: Routes traffic to an IPv4 address and some AWS resources (A: dirige el tráfico a una dirección IPv4 y algunos recursos de AWS).
7. Para Value (Valor), elija Alias para activar el recurso de alias.
8. En Route traffic to (Dirigir tráfico a), elija Alias to Cloudfront distribution (Alias a distribución de CloudFront) en el menú desplegable.
9. En el cuadro de búsqueda que dice Choose distribution (Elegir distribución), elija el nombre de dominio de la distribución de CloudFront que creó en el Paso 4.

   Para buscar el nombre de dominio de la distribución de CloudFront, haga lo siguiente:

   a. En una nueva pestaña del navegador, inicie sesión en la AWS Management Console y abra la consola de CloudFront en https://console.aws.amazon.com/cloudfront/v3/home.
   b. En el panel de navegación izquierdo, elija Distributions (Distribuciones).
   c. En la columna Origins (Orígenes), identifique la distribución de CloudFront; para ello, busque el nombre de origen, que comienza con el bucket de S3 que creó en el Paso 1 (por ejemplo, **`tutorial-bucket`**).

d. Después de encontrar la distribución de la lista, amplíe la columna Domain name (Nombre del dominio) para ver el valor del nombre del dominio para la distribución de CloudFront.

10. En la página Create record (Crear registro) en la consola de Route 53, para el resto de la configuración, conserve los valores predeterminados.
11. Elija Create records (Crear registros).

Verifique si IPv6 está habilitado para su distribución y cree otro registro de DNS si es necesario

Si IPv6 está habilitado para la distribución, debe crear otro registro de DNS.

1. Para comprobar si IPv6 está habilitado para la distribución, haga lo siguiente:

   a. Inicie sesión en la AWS Management Console y abra la consola de CloudFront en https://console.aws.amazon.com/cloudfront/v4/home.

   b. En el panel de navegación izquierdo, elija Distributions (Distribuciones).

   c. Elija el ID de la distribución de CloudFront que creó en el Paso 4.

   d. En la pestaña General, en Settings (Configuración), verifique si IPv6 está configurado como Enabled (Habilitado).

      Si IPv6 está habilitado para la distribución, debe crear otro registro de DNS.

2. Si IPv6 está habilitado para la distribución, haga lo siguiente para crear un registro de DNS:

   a. Inicie sesión en la AWS Management Console y abra la consola de Route 53 en https://console.aws.amazon.com/route53/.

   b. En el panel de navegación, elija Hosted zones (Zonas alojadas).

   c. En la página Hosted zones (Zonas alojadas), elija el nombre de la zona alojada que Route 53 creó por usted en Prerequisites (Requisitos previos) (por ejemplo, **`example.com`**).

   d. Elija Create record (Crear registro) y luego utilice el método de registro de creación rápida.

   e. Para Record name (Nombre del registro), en el campo de texto anterior al nombre de dominio personalizado, escriba el mismo valor que escribió cuando creó el registro de DNS IPv4 anterior. Por ejemplo, en este tutorial, para dirigir el tráfico del subdominio `www.example.com`, ingrese solo **`www`**.

   f. En Record type (Tipo de registro), elija A: Routes traffic to an IPv6 address and some AWS resources (A: dirige el tráfico a una dirección IPv4 y algunos recursos de AWS).

g. Para Value (Valor), elija Alias para activar el recurso de alias.

h. En Route traffic to (Dirigir tráfico a), elija Alias to Cloudfront distribution (Alias a distribución de CloudFront) en el menú desplegable.

i. En el cuadro de búsqueda que dice Choose distribution (Elegir distribución), elija el nombre de dominio de la distribución de CloudFront que creó en el Paso 4.

j. Mantenga la configuración restante establecida en los valores predeterminados.

k. Elija Crear registros.

Paso 7: acceda al video de S3 a través de la distribución CloudFront con el nombre de dominio personalizado

Para acceder al video de S3 con la URL personalizada, debe combinar su nombre de dominio alternativo con la ruta al video en el bucket de S3.

Para crear una URL personalizada para acceder al video de S3 a través de la distribución de CloudFront

1. Inicie sesión en la AWS Management Console y abra la consola de CloudFront en https://console.aws.amazon.com/cloudfront/v4/home.
2. En el panel de navegación izquierdo, elija Distributions (Distribuciones).
3. Para obtener el nombre de dominio alternativo de la distribución de CloudFront, haga lo siguiente:

   a. En la columna Origins (Orígenes), busque la distribución de CloudFront correcta; para ello, busque el nombre de origen, que comienza con el bucket de S3 que creó en el Paso 1 (por ejemplo, **`tutorial-bucket`**).

   b. Después de encontrar la distribución de la lista, amplíe la columna Alternate domain names (Nombres de dominio alternativos) para copiar el valor del nombre de dominio alternativo de la distribución de CloudFront.
4. En una nueva pestaña del navegador, pegue el nombre de dominio alternativo de la distribución de CloudFront.
5. Vuelva a la pestaña anterior del navegador y abra la consola de Amazon S3 en https://console.aws.amazon.com/s3/.
6. Encuentre la ruta a su video de S3, como se explica en el Paso 5.

7. Vuelva a la pestaña del navegador donde pegó previamente el nombre de dominio alternativo, ingrese una barra diagonal (**/**) y pegue la ruta al video de S3 (por ejemplo, `sample.mp4`).

   Ahora, su video de S3 es accesible públicamente y está alojado a través de CloudFront en una URL personalizada similar a la siguiente:

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

   Reemplace el *nombre de dominio alternativo de distribución de CloudFront* y la *ruta al video de S3* con los valores apropiados. La dirección URL de ejemplo es: **https://www.example.com/sample.mp4**

(Opcional) Paso 8: vea los datos sobre las solicitudes recibidas por su distribución de CloudFront

Para ver los datos sobre las solicitudes recibidas por la distribución de CloudFront

1. Inicie sesión en la AWS Management Console y abra la consola de CloudFront en https://console.aws.amazon.com/cloudfront/v4/home.
2. En el panel de navegación izquierdo, en Reports & analytics (Informes y análisis), elija los informes de la consola, que van desde Cache statistics (Estadísticas de caché), Popular objects (Objetos populares), Top Referrers (Principales referentes), Usage (Uso) y Viewers (Usuarios).

   Puede filtrar cada panel de informes. Para obtener más información, consulte Informes de CloudFront en la consola en la Guía para desarrolladores de Amazon CloudFront.
3. Para filtrar datos, elija el ID de la distribución de CloudFront que creó en el paso 4.

Paso 9: limpieza

Si alojó un video de streaming de S3 mediante CloudFront y Route 53 solo como ejercicio de aprendizaje, elimine los recursos de AWS que asignó para dejar de acumular cargos.

> **Note**
>
> Cuando registra un dominio, cuesta dinero inmediatamente y es irreversible. Puede elegir no renovar automáticamente el dominio, pero paga por adelantado y lo posee durante el año.

Para obtener más información, consulte Renovación de un nuevo dominio en la Guía para desarrolladores de Amazon Route 53.

Pasos secundarios

- Eliminar la distribución de CloudFront
- Eliminar el registro de DNS
- Eliminar la zona alojada pública de su dominio personalizado
- Eliminar el nombre de dominio personalizado de Route 53
- Eliminar el video original en el bucket de origen de S3
- Eliminar el bucket de origen de S3

Eliminar la distribución de CloudFront

1. Inicie sesión en la AWS Management Console y abra la consola de CloudFront en https://console.aws.amazon.com/cloudfront/v4/home.
2. En el panel de navegación izquierdo, elija Distributions (Distribuciones).
3. En la columna Origins (Orígenes), busque la distribución de CloudFront correcta; para ello, busque el nombre de origen, que comienza con el bucket de S3 que creó en el Paso 1 (por ejemplo, **`tutorial-bucket`**).
4. Antes de eliminar una distribución de CloudFront, debe desactivarla.
   - Si el valor de la columna Status (Estado) está como Enabled (Habilitado) y el valor Last modified (Última modificación) es la marca de tiempo en la que se modificó la distribución por última vez, desactive la distribución antes de eliminarla.
   - Si el valor de Status (Estado) está como Enabled (Habilitado) y el valor de Last modified (Última modificación) es Deploying (Implementación), espere hasta que el valor de Status (Estado) cambie a la marca temporal del momento en que se modificó la distribución por última vez. A continuación, siga con el paso 4 para deshabilitar la distribución antes de eliminarla.
5. Para desactivar la distribución de CloudFront, haga lo siguiente:
   a. En la lista Distributions (Distribuciones), seleccione la casilla de verificación junto al ID de la distribución que desea eliminar.

b. Para deshabilitar la distribución, elija Disable (Deshabilitar) y luego, Disable (Deshabilitar) para confirmar la operación.

   Si desactiva una distribución que tiene asociado un nombre de dominio alternativo, CloudFront deja de aceptar tráfico para ese nombre de dominio (por ejemplo: `www.example.com`), aunque haya otra distribución que tenga un nombre de dominio alternativo con un carácter comodín (`*`) que coincida con el mismo dominio (por ejemplo: `*.example.com`).

c. El valor de la columna State (Estado) cambia inmediatamente a Disabled (Deshabilitada). Espere hasta que el valor Last modified (Última modificación) cambie de Deploying (Implementación) a la marca temporal en que se modificó la distribución por última vez.

   Dado que CloudFront debe propagar este cambio a todas las ubicaciones de borde, es posible que tenga que esperar unos minutos hasta que finalice la actualización para poder eliminar la distribución.

6. Para eliminar la distribución desactivada, haga lo siguiente:

   a. Elija la casilla de verificación junto al ID de la distribución que desea eliminar.

   b. Elija Delete (Eliminar) y luego, Delete (Eliminar) para confirmar la acción.

Eliminar el registro de DNS

Si desea eliminar la zona alojada pública del dominio (incluido el registro de DNS), consulte Eliminar la zona alojada pública de su dominio personalizado en la Guía para desarrolladores de Amazon Route 53. Si solo quiere eliminar el registro de DNS creado en el Paso 6), haga lo siguiente:

1. Inicie sesión en la AWS Management Console y abra la consola de Route 53 en https://console.aws.amazon.com/route53/.
2. En el panel de navegación, elija Hosted zones (Zonas alojadas).
3. En la página Hosted zones (Zonas alojadas), elija el nombre de la zona alojada que Route 53 creó por usted en Prerequisites (Requisitos previos) (por ejemplo, **`example.com`**).
4. En la lista de registros, seleccione la casilla de verificación junto a los registros que desea eliminar (los registros que creó en el Paso 6).

> **Note**
>
> No puede eliminar los registros que tengan el valor NS o SOA en Type (Tipo).

5. Seleccione la opción Delete Record Set (Eliminar conjunto de registros).
6. Para confirmar la eliminación, elija Delete (Eliminar).

Los cambios en los registros tardan tiempo en propagarse a los servidores DNS de Route 53. En la actualidad, el único modo de verificar si los cambios se propagan es utilizar la acción de la API GetChange. Por lo general, los cambios se propagan a todos los servidores de Route 53 en un plazo de 60 segundos.

Eliminar la zona alojada pública de su dominio personalizado

> **Warning**
>
> Si desea conservar el registro del dominio, pero desea detener el enrutamiento del tráfico de Internet a su sitio web o aplicación web, se recomienda eliminar los registros en la zona alojada (como se describe en la sección anterior) en lugar de eliminar la zona alojada. Además, si elimina una zona alojada, alguien podría usar el dominio y dirigir el tráfico a sus propios recursos mediante su nombre de dominio.
> Además, la eliminación de una zona alojada es una acción que no se puede deshacer. Debe crear una nueva zona hospedada y actualizar los servidores de nombres de su registro de dominio, proceso que puede requerir hasta 48 horas en surtir efecto.
> Si desea hacer que el dominio no esté disponible en Internet, puede transferir el servicio DNS a un servicio DNS gratuito y eliminar la zona alojada de Route 53. Esto evita que las futuras consultas DNS puedan dirigirse erróneamente.
>
> 1. Si el dominio está registrado en Route 53, consulte Adición o modificación de servidores de nombres y registros de conexión de un dominio en la Guía para desarrolladores de Amazon Route 53 a fin de obtener información para reemplazar los servidores de nombres de Route 53 con los servidores de nombres del nuevo servicio DNS.
> 2. Si el dominio está registrado en otro registrador, utilice el método proporcionado por el registrador para cambiar servidores de nombres en el dominio.

> **Note**
>
> Si desea eliminar una zona hospedada para un subdominio (`www.example.com`), no es necesario cambiar servidores de nombres del dominio (`example.com`).

1. Inicie sesión en la AWS Management Console y abra la consola de Route 53 en https://console.aws.amazon.com/route53/.
2. En el panel de navegación, elija Hosted zones (Zonas alojadas).
3. En la página Hosted Zones, elija la fila de la zona hospedada que desea eliminar.
4. En la pestaña Records (Registros) de su zona alojada, confirme que la zona alojada que desea eliminar solo contiene un registro NS y otro SOA.

   Si contiene registros adicionales, elimínelos.

   Si ha creado cualquier registros NS para subdominios en la zona hospedada, elimine también esos registros.
5. En la pestaña DNSSEC signing (Firma de DNSSEC) de su zona alojada, desactive la firma de DNNSSEC si estaba habilitada. Para obtener más información, consulte Disabling DNSSEC signing (Desactivación de la firma de DNSSEC) en la Guía para desarrolladores de Amazon Route 53.
6. En la parte superior de la página de detalles de la zona alojada, elija Delete zone (Eliminar zona).
7. Ingrese **`delete`** para confirmar la eliminación y luego, elija Delete (Eliminar).

Eliminar el nombre de dominio personalizado de Route 53

Para la mayoría de los dominios de nivel superior (TLD), puede eliminar el registro si ya no lo quiere. Si elimina un registro de nombre de dominio de Route 53 antes de la fecha de vencimiento programada, AWS no le reembolsará la cuota de registro. Para obtener más información, consulte Eliminar un registro de nombre de dominio en la Guía para desarrolladores de Amazon Route 53.

> **Important**
>
> Si desea transferir el dominio entre Cuentas de AWS o transferir el dominio a otro registrador, no elimine un dominio y espere a volver a registrarlo de inmediato. En su lugar, consulte la documentación aplicable en la Guía para desarrolladores de Amazon Route 53:
>
> - Transferencia de un dominio a otra Cuenta de AWS
> - Transferencia de un dominio de Amazon Route 53 a otro registrador

Eliminar el video original en el bucket de origen de S3

1. Inicie sesión AWS Management Console Management Console y abra la consola de Amazon S3 en https://console.aws.amazon.com/s3/.
2. En el panel de navegación izquierdo, elija Instancias.
3. En la lista Bucket name (Nombre del bucket), elija el nombre del bucket al que ha cargado el video en el Paso 2 (por ejemplo: **`tutorial-bucket`**).
4. En la pestaña Objects (Objetos), seleccione la casilla de verificación junto al nombre del objeto que desea eliminar (por ejemplo, `sample.mp4`).
5. Elija Eliminar.
6. En Permanently delete objects? (¿Borrar objetos de forma permanente?), ingrese **`permanently delete`** para confirmar que desea eliminar este objeto.
7. Elija Eliminar objetos.

Eliminar el bucket de origen de S3

1. Inicie sesión AWS Management Console Management Console y abra la consola de Amazon S3 en https://console.aws.amazon.com/s3/.
2. En el panel de navegación izquierdo, elija Instancias.
3. En la lista Buckets, elija el botón de opción junto al nombre del bucket que creó en el Paso 1 (por ejemplo, **`tutorial-bucket`**).
4. Elija Eliminar.
5. En la página Delete bucket (Eliminar bucket) confirme que desea eliminar el bucket introduciendo el nombre del bucket en el campo de texto y, a continuación, elija Delete bucket (Eliminar bucket).

Siguientes pasos

Después de completar este tutorial, puede explorar más a fondo los siguientes casos de uso relacionados:

- Transcodifique los videos de S3 a los formatos de streaming necesarios para un televisor o dispositivo conectado en particular antes de alojar estos videos con una distribución de CloudFront.

  Para utilizar las operaciones por lotes de Amazon S3, AWS Lambda y AWS Elemental MediaConvert para transcodificar por lotes una recopilación de vídeos a una variedad de formatos de medios de salida, consulte Tutorial: videos de transcodificación por lotes con operaciones por lotes de S3, AWS Lambda, y AWS Elemental MediaConvert.
- Aloje otros objetos almacenados en S3, como imágenes, audio, gráficos animados, hojas de estilo, HTML, JavaScript, aplicaciones React, etc., mediante CloudFront y Route 53.

  Para ver un ejemplo, consulte Tutorial: Configuración de un sitio web estático mediante un dominio personalizado registrado con Route 53 y Aceleración de su sitio web con Amazon CloudFront.
- Use Amazon S3 Transfer Acceleration para configurar transferencias de archivos rápidas y seguras. Transfer Acceleration puede acelerar la carga de video en su bucket de S3 para transferir a larga distancia videos más grandes. Transfer Acceleration mejora el rendimiento de transferencia al enrutar el tráfico a través de las ubicaciones periféricas distribuidas globalmente de CloudFront y las redes troncales de AWS. También utiliza optimizaciones de protocolo de red. Para obtener más información, consulte Configuración de transferencias de archivos rápidas y seguras con Amazon S3 Transfer Acceleration.

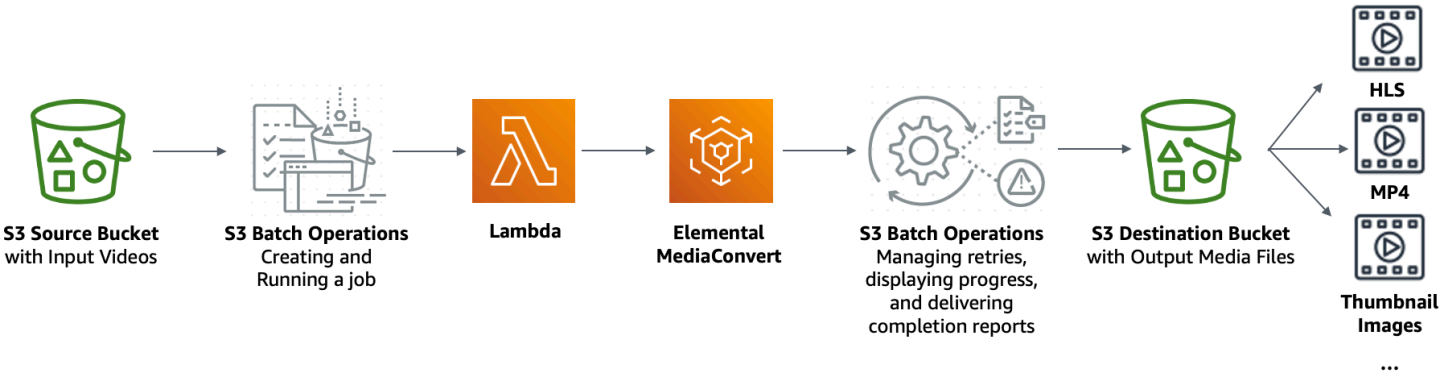
Tutorial: videos de transcodificación por lotes con operaciones por lotes de S3, AWS Lambda, y AWS Elemental MediaConvert

Los usuarios de video utilizan dispositivos de todas las formas, tamaños y cosechas para disfrutar del contenido multimedia. Esto supone un desafío para los creadores de contenido y distribuidores. En lugar de tener un formato de tamaño único, los videos deben abarcar una amplia gama de tamaños, formatos y velocidades de bits. Esta tarea de conversión es aún más desafiante cuando se cuenta con una gran cantidad de videos que se deben convertir.

AWS ofrece un método para crear una arquitectura distribuida y escalable que hace lo siguiente:

- Ingesta videos de entrada
- Procesa los videos para reproducirlos en una amplia gama de dispositivos
- Almacena los archivos multimedia transcodificados
- Entrega los archivos multimedia de salida para satisfacer la demanda

Cuando tiene repositorios de video extensos y almacenados en Amazon S3, puede transcodificar estos videos desde sus formatos fuente a varios tipos de archivo en el tamaño, la resolución o el formato necesarios para un reproductor de video o dispositivo en particular. En concreto, Operaciones por lotes de S3 le proporciona una solución para invocar funciones de AWS Lambda para los videos de entrada existentes en un bucket fuente S3. A continuación, las funciones de Lambda llaman a AWS Elemental MediaConvert para realizar tareas de transcodificación de vídeo a gran escala. Los archivos multimedia de salida convertidos se almacenan en un bucket de destino de S3.



Objetivo

En este tutorial, aprenderá a configurar operaciones por lotes de S3 para invocar una función de Lambda para la transcodificación por lotes de videos almacenados en un bucket fuente de S3. La función de Lambda llama a MediaConvert para transcodificar los videos. Las salidas para cada video en el bucket fuente de S3 son las que se muestran a continuación:

- Una transmisión de velocidad de bits adaptable HTTP Live Streaming (HLS) para la reproducción en dispositivos de varios tamaños y anchos de banda variables
- Un archivo de video MP4
- Imágenes en miniatura recopiladas a intervalos

Temas

Requisitos previos

Antes de comenzar este tutorial, necesita tener un bucket fuente de Amazon S3 (por ejemplo, **`tutorial-bucket-1`**) con videos que se transcodificarán ya almacenados en él.

Si lo desea, puede darle otro nombre al bucket. Para obtener más información acerca de los nombres de bucket de Amazon S3, consulte Reglas de nomenclatura de buckets.

Para el bucket fuente de S3, mantenga la configuración relacionada con la configuración del bloqueo del acceso público para este bucket establecida en los valores predeterminados (la opción de bloqueo de todo el acceso público está habilitada). Para obtener más información, consulte Crear un bucket.

Si quiere obtener más información para cargar videos en el bucket fuente de S3, consulte Carga de objetos. Cuando carga muchos videos grandes a S3, también puede usar Amazon S3 Transfer Acceleration para configurar transferencias de archivos rápidas y seguras. Transfer Acceleration puede acelerar la carga de video en su bucket de S3 para transferir a larga distancia videos más grandes. Para obtener más información, consulte Configuración de transferencias de archivos rápidas y seguras con Amazon S3 Transfer Acceleration.

Paso 1: Cree un bucket de S3 para los archivos multimedia

En este paso, creará un bucket de destino de S3 para almacenar archivos multimedia de salida convertidos. También se crea una configuración de uso compartido de recursos fuente cruzado

(CORS) para permitir el acceso fuente cruzado a los archivos multimedia transcodificados almacenados en el bucket de destino de S3.

Pasos secundarios

- Cree un bucket para los archivos multimedia de salida
- Agregar una configuración CORS a un bucket de salida de S3

Cree un bucket para los archivos multimedia de salida

1. Inicie sesión AWS Management Console Management Console y abra la consola de Amazon S3 en https://console.aws.amazon.com/s3/.
2. En el panel de navegación izquierdo, elija Instancias.
3. Elija Crear bucket.
4. En Bucket name (Nombre de bucket), ingrese el nombre del bucket (por ejemplo, **`tutorial-bucket-2`**).
5. En Región, elija la Región de AWS en la que desea que se encuentre el bucket.
6. Para garantizar el acceso público a los archivos multimedia de salida, en Configuración de Bloquear acceso público para este bucket, desactive Bloquear todo acceso público.

   > **Warning**
   >
   > Antes de completar este paso, consulte Bloquear el acceso público a su almacenamiento de Amazon S3 para asegurarse de que comprende y acepta los riesgos que implica otorgar el acceso público. Cuando desactiva la configuración de Block Public Access para que el bucket sea público, cualquier usuario de Internet puede acceder al bucket. Le recomendamos que bloquee todo el acceso público a sus buckets.
   > Si no desea borrar la configuración de Block Public Access, puede utilizar Amazon CloudFront para entregar los archivos multimedia transcodificados a los lectores (usuarios finales). Para obtener más información, consulte Tutorial: Alojamiento de video en streaming bajo demanda con Amazon S3, Amazon CloudFront y Amazon Route 53.

7. Seleccione la casilla de verificación junto a I acknowledge that the current settings may result in this bucket and the objects within becoming public (Reconozco que la configuración actual podría dar lugar a que este bucket y los objetos dentro se conviertan en públicos).
8. Mantenga la configuración restante establecida en los valores predeterminados.

9. Elija Crear bucket.

Agregar una configuración CORS a un bucket de salida de S3

La configuración JSON CORS define una manera para que las aplicaciones web cliente (reproductores de video en este contexto) que estén cargadas en un dominio puedan reproducir archivos multimedia de salida y transcodificados en un dominio diferente.

1. Inicie sesión AWS Management Console Management Console y abra la consola de Amazon S3 en https://console.aws.amazon.com/s3/.
2. En el panel de navegación izquierdo, elija Instancias.
3. En la lista Buckets, busque y elija el nombre del bucket que creó anteriormente (por ejemplo, **`tutorial-bucket-2`**).
4. Elija la pestaña Permisos.
5. En la sección Cross-origin resource sharing (CORS) (Compartir recursos entre orígenes [CORS]), elija Edit (Editar).
6. En el cuadro de texto de configuración CORS, copie y pegue la siguiente configuración CORS.

   La configuración de CORS debe estar en formato JSON. En este ejemplo, el atributo `AllowedOrigins` usa el carácter de comodín (`*`) para especificar todos los orígenes. Si conoce su origen específico, puede restringir el atributo `AllowedOrigins` a la URL específica de su reproductor. Para obtener más información sobre la configuración y otros atributos, consulte Elementos de una configuración de CORS.

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

    }
```

```
]
```

7. Elija Guardar cambios.

Paso 2: crear un rol de IAM para MediaConvert

Para utilizar AWS Elemental MediaConvert para transcodificar los vídeos de entrada almacenados en el bucket de S3, debe tener un rol de servicio de AWS Identity and Access Management (IAM) para otorgar permisos a MediaConvert para leer y escribir archivos de vídeo desde los buckets de origen y de destino de S3. Cuando ejecuta trabajos de transcodificación, la consola de MediaConvert utiliza este rol.

Para crear un rol de IAM para MediaConvert

1. Puede crear un rol de IAM con un nombre de rol que usted elija (por ejemplo, **`tutorial-mediaconvert-role`**). Para crear este rol, siga los pasos que se detallan en Crear el rol de MediaConvert en IAM (consola) en la Guía del usuario de AWS Elemental MediaConvert.
2. Después de crear el rol de IAM para MediaConvert, en la lista de roles, elija el nombre del rol de MediaConvert que ha creado (por ejemplo, **`tutorial-mediaconvert-role`**).
3. En la página Summary (Resumen), copie el ARN de rol, que comienza con `arn:aws:iam::`, y guarde el ARN para utilizarlo más tarde.

   Para obtener más información sobre los ARN, consulte Nombres de recursos de Amazon (ARN) en la AWS Referencia general.

Paso 3: crear un rol de IAM para su función de Lambda.

Para transcodificar videos por lotes con MediaConvert y operaciones por lotes de S3, necesita tener una función de Lambda para conectar estos dos servicios a fin de convertir videos. Esta función de Lambda necesita tener un rol de IAM que le otorgue permisos para acceder a MediaConvert y a las operaciones por lotes de S3.

Pasos secundarios

- Cree un rol de IAM para su función de Lambda
- Incruste una política en línea para el rol de IAM de su función de Lambda

Cree un rol de IAM para su función de Lambda

1. Inicie sesión en AWS Management Console Management Console y abra la consola IAM en https://console.aws.amazon.com/iam/.
2. En el panel de navegación izquierdo, elija Roles y luego, Create role (Crear rol).
3. Elija el tipo de rol del servicio de AWS y luego, en Common use cases (Casos de uso comunes), elija en Lambda.
4. Elija Siguiente: permisos.
5. En la página Attach Permissions policies (Adjuntar políticas de permisos), ingrese **`AWSLambdaBasicExecutionRole`** en el cuadro Filter policies (Filtrar políticas). Para adjuntar la política administrada AWSLambdaBasicExecutionRole a este rol a fin de otorgar permisos de escritura a Amazon CloudWatch Logs, seleccione la casilla de verificación junto a AWSLambdaBasicExecutionRole.
6. Elija Siguiente: etiquetas.
7. (Opcional) Agregue etiquetas a la política administrada.
8. Elija Next: Review (Siguiente: revisar).
9. En Role name (Nombre del rol), introduzca **`tutorial-lambda-transcode-role`**.
10. Elija Create role (Crear rol).

Incruste una política en línea para el rol de IAM de su función de Lambda

Debe utilizar una política en línea para otorgar permisos al recurso MediaConvert necesario para la ejecución de la función de Lambda.

1. Inicie sesión en AWS Management Console Management Console y abra la consola IAM en https://console.aws.amazon.com/iam/.
2. En el panel de navegación izquierdo, seleccione Roles.
3. En la lista Roles, elija el nombre del rol de IAM que ha creado anteriormente para la función de Lambda (por ejemplo, **`tutorial-lambda-transcode-role`**).
4. Elija la pestaña Permisos.
5. Elija Add inline policy (Agregar política insertada).
6. Elija la pestaña JSON y luego, copie y pegue la siguiente política JSON.

Reemplace el valor ARN de ejemplo de `Resource` en la política JSON con el ARN del rol de IAM para MediaConvert que creó en el Paso 2 (por ejemplo, **`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. Elija Review Policy (Revisar la política).
8. En Name (Nombre), ingrese **`tutorial-lambda-policy`**.
9. Seleccione Crear política.

   Una vez que cree una política en línea, se integra automáticamente en el rol de IAM de su función de Lambda.

Paso 4: Cree una función de Lambda para la transcodificación de video

En esta sección del tutorial, crea una función de Lambda con el SDK para Python a fin de integrarse con las operaciones por lote de S3 y MediaConvert. Para comenzar a transcodificar los videos ya almacenados en el bucket fuente de S3, debe ejecutar un trabajo de operaciones por lote de S3 que invoca directamente la función de Lambda para cada video del bucket fuente de S3. A continuación, la función de Lambda envía un trabajo de transcodificación para cada video a MediaConvert.

Pasos secundarios

- Escriba el código de la función de Lambda y cree un paquete de implementación
- Crear una función de Lambda con un rol de ejecución (consola)
- Implemente su función de Lambda con archivos.zip y configure la función de Lambda (consola)

Escriba el código de la función de Lambda y cree un paquete de implementación

1. En el equipo local, cree un archivo con el nombre `batch-transcode`.
2. En la carpeta `batch-transcode`, cree un archivo con la configuración de trabajo de JSON. Por ejemplo, puede utilizar la configuración que se proporcionó en esta sección y nombrar al archivo `job.json`.

   Un archivo `job.json` especifica lo siguiente:

   - Qué archivos transcodificar

- Cómo quiere transcodificar sus videos de entrada
- Qué archivos multimedia de salida desea crear
- Cómo nombrar a los archivos transcodificados
- Dónde guardar los archivos transcodificados
- Qué características avanzadas aplicar, etc.

En este tutorial, utilizamos el archivo `job.json` para crear las siguientes salidas para cada video en el bucket fuente de S3:

- Una transmisión de velocidad de bits adaptable HTTP Live Streaming (HLS) para la reproducción en dispositivos de diferentes tamaños y con varias bandas anchas
- Un archivo de video MP4
- Imágenes en miniatura recopiladas a intervalos

Este archivo `job.json` de ejemplo utiliza la velocidad de bits variable definida en función de la calidad (QVCR) para optimizar la calidad del video. La salida HLS es compatible con Apple (audio sin mezcla de video, duración de segmento adecuada de 6 segundos y calidad de video optimizada a través de QVBR automático).

Si no desea utilizar la configuración de ejemplo que se proporciona aquí, puede generar una especificación de `job.json` según su caso de uso. Para garantizar la coherencia entre las salidas, asegúrese de que los archivos de entrada tienen configuraciones de video y audio similares. Cree automatizaciones separadas (única configuración de `job.json`) para cualquier archivo de entrada con diferentes configuraciones de video y audio. Para obtener más información, consulte Ejemplos de configuración de tareas de AWS Elemental MediaConvert en JSON en la Guía del usuario de 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. En la carpeta `batch-transcode`, cree un archivo con una función de Lambda. Puede usar el siguiente ejemplo de Python y nombrar el archivo`convert.py`.

   Operaciones por lotes de S3 envía datos específicos de tareas a una función de Lambda y requiere datos de resultados de vuelta. Para obtener ejemplos de solicitud y respuesta para la función de Lambda, información sobre códigos de respuesta y resultado, y ejemplos de funciones de Lambda para operaciones por lote de S3, consulte Invocar a la función AWS Lambda.

```
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. Para crear un paquete de implementación con `convert.py` y `job.json` como un archivo `.zip` llamado `lambda.zip`, en su terminal local, abra la carpeta `batch-transcode` que creó anteriormente y ejecute el comando siguiente.

   Para usuarios de macOS, ejecute el siguiente comando:

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

   Para usuarios de Windows, ejecute los siguientes comandos:

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

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

Crear una función de Lambda con un rol de ejecución (consola)

1. Abra la consola de AWS Lambda en https://console.aws.amazon.com/lambda/.
2. En el panel de navegación izquierdo, elija Instancias.
3. Elija Create function (Crear función).
4. Elija Author from scratch (Crear desde cero).
5. Bajo Basic information (Información básica), haga lo siguiente:
   a. En Function name (Nombre de la función), introduzca **`tutorial-lambda-convert`**.
   b. Para Runtime (Tiempo de ejecución), elija Python 3.8 o una versión posterior.
6. SeleccionarCambiar la función de ejecución predeterminadaEnRol de ejecución, elijaUso de una función existente.
7. En Existing role (Rol existente), elija el nombre del rol de IAM que creó para su función de Lambda en el Paso 3 (por ejemplo, **`tutorial-lambda-transcode-role`**).
8. Mantenga la configuración restante establecida en los valores predeterminados.
9. Elija Crear función.

Implemente su función de Lambda con archivos.zip y configure la función de Lambda (consola)

1. En la sección Code source (Código fuente) de la página para la función de Lambda que creó anteriormente (por ejemplo, **`tutorial-lambda-convert`**), elija Upload from (Cargar desde) y luego, el archivo .zip.
2. Seleccione Upload (Cargar) para seleccionar el archivo `.zip` local.
3. Elija el archivo `lambda.zip` que creó anteriormente y luego, Open (Abrir).
4. Seleccione Guardar.
5. En la sección Runtime settings (Configuración de tiempo de ejecución), elija Edit (Editar).
6. Para indicarle al tiempo de ejecución de Lambda qué método de controlador en su código de función de Lambda invocar, ingrese **`convert.handler`** en el campo Handler (Controlador).

Al configurar una función en Python, el valor de la configuración del controlador es el nombre del archivo y el nombre de un módulo del controlador exportado, separados por un punto (`.`). Por ejemplo, `convert.handler` llama al método `handler` definido en `convert.py`.

7. Seleccione Guardar.
8. En la página de la función de Lambda, seleccione la opciónConfiguraciónPestaña. En el panel de navegación izquierdo de la pestaña Configuration (Configuración), elija Environment variables (Variables de entorno) y luego, Edit (Editar).
9. Elija Add environment variable (Añadir variable de entorno). Después ingrese la clave y el valor para cada una de las siguientes variables de entorno:

   - Clave: **`DestinationBucket`** Valor: **`tutorial-bucket-2`**

     Este valor es el bucket de S3 para los archivos multimedia de salida que creó en el Paso 1.
   - Clave: **`MediaConvertRole`** Valor: **`arn:aws:iam::`*`111122223333`*`:role/tutorial-mediaconvert-role`**

     Este valor es el ARN del rol de IAM para MediaConvert que creó en el Paso 2. Asegúrese de reemplazar este ARN con el ARN real de su rol de IAM.
   - Clave: **`Application`** Valor: **`Batch-Transcoding`**

     Este valor es el nombre de la aplicación.
10. Seleccione Guardar.
11. (Opcional) En elConfiguración, en la pestañaConfiguración generalEn el panel de navegación izquierdo, elijaEditar. En el campo Timeout (Tiempo de espera), escriba **`2`** min **`0`** sec. A continuación, elija Save (Guardar).

    Timeout (Tiempo de espera): período durante el cual Lambda permite que se ejecute una función antes de pararla. El valor predeterminado es de 3 segundos. Los precios se basan en la cantidad de memoria configurada y en la cantidad de tiempo durante la que se ejecuta el código. Para más información, consulte Precios de AWS Lambda.

Paso 5: Configure un inventario de Amazon S3 para un bucket fuente de S3

Después de configurar la función de Lambda de transcodificación, cree un trabajo de operaciones por lote de S3 para transcodificar un conjunto de videos. En primer lugar, necesita una lista de los objetos de video de entrada en los que quiere que Operaciones Batch de S3 ejecute la acción de

transcodificación especificada. Para obtener una lista de objetos de video de entrada, puede generar un informe de inventario de S3 para su bucket fuente de S3 (por ejemplo, **`tutorial-bucket-1`**).

Pasos secundarios

- Cree y configure un bucket para los informes de inventario de S3 de videos de entrada
- Configuración del inventario de Amazon S3 para un bucket fuente de video de S3
- Verifique el informe de inventario para su bucket fuente de video de S3

Cree y configure un bucket para los informes de inventario de S3 de videos de entrada

Para almacenar informes de inventario de S3 que muestran los objetos del bucket fuente de S3, debe crear un bucket de destino de inventario de S3 y configurar una política de bucket para que el bucket escriba archivos de inventario en el bucket fuente de S3.

1. Inicie sesión AWS Management Console Management Console y abra la consola de Amazon S3 en https://console.aws.amazon.com/s3/.
2. En el panel de navegación izquierdo, elija Instancias.
3. Elija Crear bucket.
4. En Bucket name (Nombre de bucket), ingrese el nombre del bucket (por ejemplo, **`tutorial-bucket-3`**).
5. En Región de AWS, elija la Región de AWS en la que desea que se encuentre el bucket.

   El bucket de destino del inventario debe estar en la misma Región de AWS que el bucket fuente en el que configura el inventario de S3. El bucket de destino del inventario puede estar en una Cuenta de AWS diferente.
6. En la configuración de Block Public Access para este bucket, mantenga la configuración predeterminada (la opción Block all public access [Bloquear todo el acceso público] está habilitada).
7. Mantenga la configuración restante establecida en los valores predeterminados.
8. Elija Crear bucket.
9. En la lista de Buckets, busque y seleccione el nombre del bucket de que creó anteriormente (por ejemplo: **`tutorial-bucket-3`**).
10. Para otorgar permiso a Amazon S3 para escribir los datos de los informes de inventario en el bucket de destino de inventario de S3, elija la pestaña Permissions (Permisos).

11. Desplácese hacia abajo hasta la sección Bucket policy (Política de bucket) y luego, elija Edit (Editar). Se abre la página de la política de bucket.
12. Para otorgar permisos para el inventario de S3, en el campo Policy (Política), pegue la siguiente política de bucket.

    Reemplace los tres valores de ejemplo por los siguientes valores:

    - El nombre del bucket que ha creado para almacenar los informes de inventario (por ejemplo:*tutorial-bucket-3*).
    - El nombre del bucket fuente que almacena los videos de entrada (por ejemplo, *tutorial-bucket-1*).
    - El ID de la Cuenta de AWS que utilizó para crear el bucket fuente de video de S3 (por ejemplo, *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. Elija Guardar cambios.

Configuración del inventario de Amazon S3 para un bucket fuente de video de S3

Debe configurar el inventario de S3 para generar una lista de archivos planos de objetos de video y metadatos para su bucket fuente de video S3. Estos informes programados de inventario pueden incluir todos los objetos del bucket u objetos agrupados por un prefijo compartido. En este tutorial, el informe de inventario de S3 incluye todos los objetos de video del bucket fuente de S3.

1. Inicie sesión AWS Management Console Management Console y abra la consola de Amazon S3 en https://console.aws.amazon.com/s3/.
2. En el panel de navegación izquierdo, elija Instancias.
3. Para configurar un informe de inventario de S3 de los videos de entrada en el bucket fuente de S3, en la lista Buckets, elija el nombre del bucket fuente de S3 (por ejemplo, **`tutorial-bucket-1`**).
4. Seleccione la pestaña Management.
5. Desplácese hasta la sección Inventory configurations (Configuraciones de inventario) y elija Create inventory configuration (Creación de la configuración de inventario).
6. ParaNombre de la configuración del inventario, escriba un nombre (por ejemplo:**`tutorial-inventory-config`**).
7. En Inventory scope (Alcance del inventario), elija Current version only (Versión actual solamente) para Object versions (Versiones de objetos) y mantenga la otra configuración de Inventory scope (Alcance del inventario) establecida en los valores predeterminados de este tutorial.
8. En la sección Report details (Datos del informe), para Destination bucket (Bucket de destino), elija This account (Esta cuenta).
9. Para Destination (Destino), elija Browse SE (Examinar S3) y el bucket de destino que creó anteriormente para guardar los informes de inventario (por ejemplo,**`tutorial-bucket-3`**). Luego, elija Choose path (Elegir ruta).

   El bucket de destino del inventario debe estar en la misma Región de AWS que el bucket fuente en el que configura el inventario de S3. El bucket de destino del inventario puede estar en una Cuenta de AWS diferente.

   En el campo para bucket Destination (Destino) verá el permiso de bucket de destino que se agrega a la política de bucket de destino para permitir que Amazon S3 coloque datos en ese bucket. Para obtener más información, consulte Creación de una política de bucket de destino.
10. Para Frequency (Frecuencia), elija Daily (Diario).
11. Elija CSV para Output format (Formato de salida).

12. Para Status (Estado), elija Enabled (Habilitado).
13. En la sección Server-side encryption (Cifrado en el servidor), elija Disable (Deshabilitar) para este tutorial.

    Para obtener más información, consulte Configuración del inventario mediante la consola de S3 y Concesión de permiso a Amazon S3 con el fin de utilizar su clave administrada por el cliente para el cifrado.
14. En la sección Additional fields - optional (Campos adicionales: opcional), seleccione Size (Tamaño), Last modified (Última modificación) y Storage class (Clase de almacenamiento).
15. Seleccione Crear.

Para obtener más información, consulte Configuración del inventario mediante la consola de S3.

Verifique el informe de inventario para su bucket fuente de video de S3

Cuando se publica un informe de inventario, los archivos de manifiesto se envían al bucket de destino del inventario de S3.

1. Inicie sesión AWS Management Console Management Console y abra la consola de Amazon S3 en https://console.aws.amazon.com/s3/.
2. En el panel de navegación izquierdo, elija Instancias.
3. En la lista Buckets, elija el nombre del bucket de código fuente de video (por ejemplo: **`tutorial-bucket-1`**).
4. Elija Management (Administración).
5. Para ver si el informe de inventario de S3 está listo para crear un trabajo de operaciones por lote de S3 en el Paso 7, en Inventory configurations (Configuración del inventario), verifique si el botón Create job from manifest (Crear trabajo a partir del manifiesto) está habilitado.

> **Note**
>
> Se pueden tardar hasta 48 horas en entregar el primer informe. Si el archivo deCrear trabajo a partir del manifiestoestá desactivado, no se ha entregado el primer informe de inventario. Espere hasta que se entregue el primer informe de inventario y se habilite el botón Create job from manifest (Crear trabajo a partir del manifiesto) para crear un trabajo de operaciones por lote de S3 en el Paso 7.

6. Para verificar un informe de inventario de S3 (`manifest.json`), en la columna Destination (Destino), elija el nombre del bucket de destino de inventario que creó anteriormente para almacenar informes de inventario (por ejemplo, **`tutorial-bucket-3`**).
7. En la pestaña Objects (Objetos), elija la carpeta existente con el nombre del bucket fuente de S3 (por ejemplo, **`tutorial-bucket-1`**). Luego elija el nombre que ingresó en Inventory configuration name (Nombre de la configuración del inventario) cuando creó la configuración de inventario (por ejemplo, **`tutorial-inventory-config`**).

   Puede ver una lista de carpetas con las fechas de generación de los informes como sus nombres.
8. Para verificar el informe diario de inventario de S3 en una fecha, elija una carpeta con un nombre de fecha de generación y luego, elija `manifest.json`.
9. Para comprobar los detalles del informe de inventario en una fecha específica, en la página manifest.json, elija Descargar o Abrir.

Paso 6: creación de un rol de IAM para Operaciones por Batch de S3

Para utilizar operaciones por lote de S3 para realizar la transcodificación por lote, primero debe crear un rol de IAM a fin de otorgar permisos a Amazon S3 para realizar operaciones por lote de S3.

Pasos secundarios

- Creación de una política de IAM para Operaciones Batch de S3
- Cree un rol de IAM y asigne permisos de Operaciones por lotes de S3 para su ejecución.

Creación de una política de IAM para Operaciones Batch de S3

Debe crear una política de IAM que otorgue permiso a las operaciones por lote de S3 para leer el manifiesto de entrada, invocar a la función de Lambda y escribir el informe de finalización de trabajos de operaciones por lote de S3.

1. Inicie sesión en AWS Management Console Management Console y abra la consola IAM en https://console.aws.amazon.com/iam/.
2. En el panel de navegación izquierdo, elija Policies (Políticas).
3. Elija Create Policy (Crear política).
4. Seleccione la pestaña JSON.

5. Copie la siguiente política de confianza en el campo de texto JSON.

   En la política JSON, sustituya los cuatro valores de ejemplo por los siguientes:

   - El nombre del bucket fuente que almacena los videos de entrada (por ejemplo, *tutorial-bucket-1*).
   - El nombre del bucket de destino de inventario que ha creado en el Paso 5 para almacenar archivos `manifest.json` (por ejemplo: *tutorial-bucket-3*).
   - El nombre del bucket que creó en el Paso 1 para almacenar archivos multimedia de salida (por ejemplo, *tutorial-bucket-2*). En este tutorial, ponemos los informes de finalización de trabajos en el bucket de destino para los archivos de medios de salida.
   - El rol ARN de la función de Lambda que ha creado en el Paso 4. Para buscar y copiar el ARN de rol de la función de Lambda, haga lo siguiente:
     - En una nueva pestaña del navegador, abra la página Functions (Funciones) en la consola de Lambda en https://console.aws.amazon.com/lambda/home#/functions.
     - En la lista Functions (Funciones), elija el nombre de la función de Lambda que creó en el Paso 4 (por ejemplo, **tutorial-lambda-convert**).
     - Seleccionar Copy ARN (Copiar 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. Elija Siguiente: Etiquetas.
7. Elija Siguiente: Revisar.
8. En el campo Name (Nombre), escriba **tutorial-s3batch-policy**.
9. Elija Crear política.

Cree un rol de IAM y asigne permisos de Operaciones por lotes de S3 para su ejecución.

1. Inicie sesión en AWS Management Console Management Console y abra la consola IAM en https://console.aws.amazon.com/iam/.
2. En el panel de navegación izquierdo, elija Roles y luego, Create role (Crear rol).
3. Elija el tipo de rol de Servicio de AWS y luego, el servicio de S3.
4. En Select your use case (Seleccione su caso de uso), elija S3 Batch Operations (Operaciones por lotes de S3).
5. Elija Siguiente: permisos.
6. En Attach permissions policies (Adjuntar políticas de permisos), ingrese el nombre de la política de IAM que creó anteriormente (por ejemplo, **tutorial-s3batch-policy**) en el cuadro de búsqueda para filtrar la lista de políticas. Seleccione la casilla de verificación junto al nombre de la política (por ejemplo, **tutorial-s3batch-policy**).
7. Elija Siguiente: Etiquetas.
8. Elija Next: Review (Siguiente: revisar).

9. En Role name (Nombre del rol), introduzca **`tutorial-s3batch-role`**.
10. Elija Create role (Crear rol).

    Después de crear el rol de IAM para las operaciones por lote de S3, la siguiente política de confianza se adjunta automáticamente al rol. La política de confianza permite que la entidad del servicio de las operaciones por lote de S3 pueda asumir el rol de IAM.

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

Paso 7: configurar y ejecutar el trabajo de la herramienta de operaciones por lotes de S3

Para crear un trabajo de operaciones por lote de S3 a fin de procesar los videos de entrada en el bucket fuente de S3, debe especificar parámetros para este trabajo concreto.

> **Note**
>
> Para comenzar a crear un trabajo de operacones por lote de S3, debe asegurarse de que el botón Create job from manifest (Crear trabajo a partir del manifiesto) está habilitado. Para obtener más información, consulte Verifique el informe de inventario para su bucket fuente de video de S3. Si el botón Create job from manifest (Crear trabajo a partir del manifiesto) está desactivado, el primer informe de inventario no se entrega y tiene que esperar hasta que el botón se habilite. Después de configurar el inventario de Amazon S3 para el bucket fuente de S3 en el Paso 5, puede tardar hasta 48 horas en entregar el primer informe de inventario.

Pasos secundarios

- Creación de un trabajo de Operaciones por lotes de S3
- Ejecute el trabajo de Operaciones Batch de S3 para invocar a su función de Lambda
- (Opcional) Verificación del informe de finalización
- (Opcional) Supervisar cada invocación de Lambda en la consola Lambda
- (Opcional) Supervise cada trabajo de transcodificación de video de MediaConvert en la consola de MediaConvert

Creación de un trabajo de Operaciones por lotes de S3

1. Inicie sesión AWS Management Console Management Console y abra la consola de Amazon S3 en https://console.aws.amazon.com/s3/.
2. En el panel de navegación izquierdo, elija Batch Operations (Operaciones por lote).
3. Seleccione Crear trabajo.
4. Para Región de AWS, elija la región en la que desea crear el trabajo.

   En este tutorial, para utilizar el trabajo de operaciones por lote de S3 para invocar una función de Lambda, debe crear el trabajo en la misma región que el bucket fuente de video de S3 donde se encuentran los objetos a los que se hace referencia en el manifiesto.
5. En la sección Manifest (Manifiesto), haga lo siguiente:

   a. En Manifest format (Formato del manifiesto), elija S3 Inventory report (manifest.json) (Informe de S3 Inventory [manifest.json]).

   b. Para Manifest object (Objeto del manifiesto), elija Browse S3 (Examinar S3) a fin de encontrar el bucket que creó en el Paso 5para almacenar informes de inventario (por ejemplo, **`tutorial-bucket-3`**). En la página Manifest object (Objeto del manifiesto), navegue por los nombres de los objetos hasta encontrar un archivo `manifest.json` para una fecha específica. Este archivo enumera la información sobre todos los videos que desea transcodificar por lotes. Cuando encuentre el archivo `manifest.json` que desea utilizar, elija el botón de opción situado junto a él. Luego, elija Choose path (Elegir ruta).

   c. (Opcional) Para Manifest object version ID - optional (ID de versión del objeto de manifiesto: opcional), ingrese el ID de versión del objeto de manifiesto si desea utilizar otra versión que no sea la más reciente.
6. Elija Siguiente.

7. Para utilizar la función de Lambda para transcodificar todos los objetos enumerados en el archivo `manifest.json` en Operation type (Tipo de operación), elija Invoke AWS Lambda function (Invocar función).
8. En la sección Invoke Lambda function (Invocar función de Lambda), haga lo siguiente:

   a. SeleccionarElija una de las funciones de su cuenta.

   b. Para Lambda function (Función de Lambda), elija la función de Lambda que creó en el Paso 4 (por ejemplo, **`tutorial-lambda-convert`**).

   c. Para Lambda function version (Versión de función de Lambda), conserve el valor predeterminado $LATEST.
9. Elija Siguiente. Se abre la página Configure additional options (Configurar opciones adicionales).
10. En Additional options (Opciones adicionales), conserve la configuración predeterminada.

    Para obtener más información sobre estas opciones, consulte Elementos de una solicitud de trabajo de Operaciones por lotes.
11. En la sección Completion report (Informe de finalización), para Path to completion report destination (Ruta al destino del informe de finalización), elija Browse S3 (Examinar S3). Busque el bucket que creó en el Paso 1 para los archivos multimedia de salida (por ejemplo, **`tutorial-bucket-2`**). Elija el botón de opción junto al nombre de ese bucket. Luego, elija Choose path (Elegir ruta).

    Para la configuración restante Completion report (informe de finalización), mantenga los valores predeterminados. Para obtener más información sobre cómo configurar los informes, consulte Elementos de una solicitud de trabajo de Operaciones por lotes. Un informe de finalización mantiene un registro de los detalles del trabajo y las operaciones realizadas.
12. En la sección Permissions (Permisos), elija Choose from existing IAM roles (Elegir entre los roles de IAM existentes). Para Rol de IAM, elija el rol de IAM para el trabajo de Operaciones Batch de S3 que ha creado en el Paso 6 (por ejemplo: **`tutorial-s3batch-role`**).
13. Elija Siguiente.
14. En la página Review (Revisar), revise la configuración. Después elija Create job (Crear trabajo).

    Después de que S3 termina de leer el manifiesto del trabajo de operaciones por lote de S3, configura el estado del trabajo en Awaiting your confirmation to run (En espera de su confirmación para ejecutarse). Para ver las actualizaciones del estado del trabajo, actualice la página. No puede ejecutar el trabajo hasta que el estado sea Awaiting your confirmation to run (En espera de su confirmación para ejecutarse).

Ejecute el trabajo de Operaciones Batch de S3 para invocar a su función de Lambda

Ejecute su trabajo de operaciones Batch para invocar su función de Lambda para la transcodificación de video. Si el trabajo falla, puede verifica el informe de finalización para identificar la causa.

Para ejecutar el trabajo de operaciones Batch de S3

1. Inicie sesión AWS Management Console Management Console y abra la consola de Amazon S3 en https://console.aws.amazon.com/s3/.
2. En el panel de navegación izquierdo, elija Batch Operations (Operaciones por lote).
3. En la lista Jobs (Trabajos), elija la opción Job ID (ID de trabajo)del trabajo en la primera fila, que es el trabajo de operaciones por lote de S3 que creó anteriormente.
4. Elija Run job (Ejecutar trabajo).
5. Revise los parámetros de trabajo de nuevo y confirme que el valor deTotal de objetos enumerados en el manifiestoEs el mismo que el número de objetos en el manifiesto. Después elija Run job (Ejecutar trabajo).

   Se abrirá la página de trabajos de Operaciones Batch de S3.
6. Después de que el trabajo comience a ejecutarse, en la página de trabajo, enEstado, compruebe el progreso de su trabajo de Operaciones Batch de S3, comoEstado,% completo,Total correcto (tasa),Total fallado (tasa),Fecha de finalización, yMotivo de la terminación.

   Cuando finalice el trabajo de operaciones por lote de S3, vea los datos en la página de trabajo para confirmar que el trabajo se haya completado como se esperaba.

   Si más del 50 por ciento de las operaciones de objetos de un trabajo de operaciones por lote de S3 falla después de que se hayan intentado más de 1000 operaciones, el trabajo falla automáticamente. Para verificar el informe de finalización para identificar la causa de los errores, consulte el procedimiento opcional que aparece a continuación.

(Opcional) Verificación del informe de finalización

Puede utilizar el informe de finalización para determinar qué objetos han fallado y la causa de los errores.

Para verificar el informe completo en busca de detalles acerca de los objetos con errores

1. En la página de su trabajo de operaciones por lote de S3, en la sección Completion report (Informe de finalización), elija el enlace de Completion report destination (Destino del informe de finalización).

   Se abre la página del bucket de destino de salida de S3.
2. En la pestaña Objects (Objetos), elija la carpeta que tiene un nombre que termina con el ID de trabajo del trabajo de operaciones por lote de S3 que creó anteriormente.
3. SeleccionarResultados/.
4. Seleccione la casilla de verificación situada junto al grupo de `.csv`.
5. Para ver el informe de trabajo, elija Open (Abrir) o Download (Descargar).

(Opcional) Supervisar cada invocación de Lambda en la consola Lambda

Después de que el trabajo de operaciones por lote de S3 comienza a ejecutarse, el trabajo invoca la función de Lambda para cada objeto de video de entrada. S3 escribe registros de cada invocación de Lambda en CloudWatch Logs. Puede utilizar el panel de monitoreo de la consola de Lambda para monitorear sus funciones y aplicaciones de Lambda.

1. Abra la consola de AWS Lambda en https://console.aws.amazon.com/lambda/.
2. En el panel de navegación izquierdo, elija Instancias.
3. En la lista Functions (Funciones), elija el nombre de la función de Lambda que creó en el Paso 4 (por ejemplo, **`tutorial-lambda-convert`**).
4. Elija la pestaña Monitor (Monitorear).
5. En Metrics (Métricas), consulte las métricas de tiempo de ejecución de su función de Lambda.
6. En Logs (Registros), vea los datos de registro de cada invocación de Lambda a través de CloudWatch Logs Insights.

> **Note**
>
> Cuando se utiliza S3 Batch Operations con una función de Lambda, se invoca la función de Lambda en cada objeto. Si su trabajo de operaciones Batch de S3 es grande, puede invocar varias funciones de Lambda al mismo tiempo, causando un aumento en la concurrencia de Lambda.

Cada Cuenta de AWS tiene una cuota de concurrencia de Lambda por región. Para obtener más información, consulte el escalado de funciones de AWS Lambda en la Guía para desarrolladores de AWS Lambda. Una práctica recomendada para usar funciones Lambda con Operaciones Batch de S3 es establecer un límite de concurrencia en la propia función de Lambda. Esto evita que su trabajo consuma la mayor parte de su concurrencia de Lambda y potencialmente limita otras funciones de su cuenta. Para obtener más información, consulte Managing Lambda reserved concurrency (Gestión de la simultaneidad reservada de Lambda) en la Guía para desarrolladores de AWS Lambda.

(Opcional) Supervise cada trabajo de transcodificación de video de MediaConvert en la consola de MediaConvert

Una tarea de MediaConvert se encarga de transcodificar un archivo multimedia. Cuando su trabajo de operaciones por lote de S3 invoca la función de Lambda para cada video, cada invocación de Lambda crea un trabajo de transcodificación de MediaConvert para cada video de entrada.

1. Inicie sesión en laAWS Management Consoley abra la consola de MediaConvert enhttps://console.aws.amazon.com/mediaconvert/.
2. Si aparece la página de introducción a MediaConvert, elija Get Started.
3. En la lista deTrabajos, vea cada fila para supervisar la tarea de transcodificación de cada video de entrada.
4. Identifique la fila de un trabajo que desea verificar y elija el enlace Job ID (ID de trabajo) para abrir la página de detalles del trabajo.
5. En la páginaResumen del Jobpágina, enSalidas, elija el enlace para la salida HLS, MP4 o Thumbnails, dependiendo de lo que admita su navegador, para ir al bucket de destino S3 para los archivos multimedia de salida.
6. En la carpeta correspondiente (HLS, MP4 o Thumbnails) del bucket de destino de salida S3, elija el nombre del objeto de archivo de medios de salida.

   Se abre la página de detalles del objeto.
7. En la página de objetos, en Object overview (Información general del objeto), elija el enlace de URL del objeto para ver el archivo multimedia de salida transcodificado.

Paso 8: Compruebe los archivos multimedia de salida desde su bucket de destino S3

Para verificar los archivos multimedia de salida desde el bucket de destino de S3

1. Inicie sesión AWS Management Console Management Console y abra la consola de Amazon S3 en https://console.aws.amazon.com/s3/.
2. En el panel de navegación izquierdo, elija Instancias.
3. En la lista Buckets, elija el nombre del bucket de destino de S3 para los archivos multimedia de salida que creó en el Paso 1 (por ejemplo, **`tutorial-bucket-2`**).
4. En la páginaObjetos, cada video de entrada tiene una carpeta con el nombre del video de entrada. Cada carpeta contiene los archivos de medios de salida transcodificados para un video de entrada.

   Para comprobar los archivos de medios de salida en busca de un video de entrada, haga lo siguiente:

   a. Elija la carpeta con el nombre del video de entrada que desea verificar.
   b. Elija la carpeta Default/.
   c. Elija la carpeta para un formato transcodificado (HLS, MP4 o miniaturas en este tutorial).
   d. Elija el nombre del archivo de medios de salida.
   e. Para ver el archivo transcodificado, en la página de detalles del objeto, elija el enlace situado en Object URL (URL del objeto).

      Los archivos de medios de salida en formato HLS se dividen en segmentos cortos. Para reproducir estos videos, debe incrustar la URL del objeto del archivo `.m3u8` en un reproductor compatible.

Paso 9: limpiar

Si transcodificó videos con Operaciones por lotes de S3, Lambda y MediaConvert solo como ejercicio de aprendizaje, elimine los recursos AWS que asignó para que ya no acumule cargos.

Pasos secundarios

- Elimine la configuración del inventario de S3 para el bucket de código fuente de S3
- Elimine la función de Lambda

- Eliminación del grupo de registros de CloudWatch
- Eliminar los roles de IAM junto con las directivas en línea para los roles de IAM
- Elimine la política de IAM administrada por el cliente
- Vacíe los buckets de S3
- Eliminación del bucket de S3

Elimine la configuración del inventario de S3 para el bucket de código fuente de S3

1. Inicie sesión AWS Management Console Management Console y abra la consola de Amazon S3 en https://console.aws.amazon.com/s3/.
2. En el panel de navegación izquierdo, elija Instancias.
3. En la lista Buckets, elija el nombre del bucket fuente (por ejemplo: **`tutorial-bucket-1`**).
4. Seleccione la pestaña Management.
5. En la sección Inventory configurations (Configuración del inventario), elija la configuración de inventario que creó en el Paso 5 (por ejemplo, **`tutorial-inventory-config`**).
6. Elija Delete (Eliminar) y, a continuación, Confirm (Confirmar).

Elimine la función de Lambda

1. Abra la consola de AWS Lambda en https://console.aws.amazon.com/lambda/.
2. En el panel de navegación izquierdo, elija Instancias.
3. Seleccione la casilla de verificación junto a la función que creó en el Paso 4 (por ejemplo, **`tutorial-lambda-convert`**).
4. Elija Acciones y, a continuación, elija Eliminar.
5. En el cuadro de diálogo Eliminar función, elija Eliminar.

Eliminación del grupo de registros de CloudWatch

1. Abra la consola de CloudWatch en https://console.aws.amazon.com/cloudwatch/.
2. En el panel de navegación izquierdo, elija Logs (Registros) y luego, Log groups (Grupos de registros).

3. Seleccione la casilla de verificación junto al grupo de registros que tiene un nombre que termina con la función de Lambda que ha creó en el Paso 4 (por ejemplo, **`tutorial-lambda-convert`**).
4. Elija Actions (Acciones) y, a continuación, elija Delete log group (Eliminar grupo de registro).
5. En el cuadro de diálogo Delete log group(s), Eliminar grupo(s) de registro(s) elija Delete (Eliminar).

Eliminar los roles de IAM junto con las directivas en línea para los roles de IAM

Para eliminar los roles de IAM que creó en el Paso 2, Paso 3, y Paso 6, realice una de las siguientes opciones:

1. Inicie sesión en AWS Management Console Management Console y abra la consola IAM en https://console.aws.amazon.com/iam/.
2. En el panel de navegación izquierdo, elija Roles y luego, seleccione las casillas de verificación junto a los nombres de rol que desee eliminar.
3. En la parte superior de la página, elija Delete (Eliminar).
4. En el cuadro de diálogo de confirmación, escriba la respuesta necesaria en el campo de entrada de texto basado en la solicitud de datos y elija Eliminar.

Elimine la política de IAM administrada por el cliente

Para eliminar la política de IAM administrada por el cliente que creó en el Paso 6, realice lo siguiente:

1. Inicie sesión en AWS Management Console Management Console y abra la consola IAM en https://console.aws.amazon.com/iam/.
2. En el panel de navegación izquierdo, elija Policies (Políticas).
3. Elija el botón de opción situado junto a la política que creó en el Paso 6 (por ejemplo, **`tutorial-s3batch-policy`**). Puede utilizar el cuadro de búsqueda para filtrar la lista de políticas.
4. Elija Acciones y, a continuación, elija Eliminar.
5. Confirme que desea eliminar esta política; para ello, ingrese su nombre en el campo de texto y elija Delete (Eliminar).

Vacíe los buckets de S3

Para vaciar los buckets de S3 que creó en Prerequisites (Requisitos previos), Paso 1 y Paso 5, realice lo siguiente:

1. Inicie sesión AWS Management Console Management Console y abra la consola de Amazon S3 en https://console.aws.amazon.com/s3/.
2. En el panel de navegación izquierdo, elija Instancias.
3. En la lista Buckets, elija el botón de opción del bucket junto al nombre del bucket que desea vaciar y luego, Empty (Vaciar).
4. En la página Empty bucket (Vaciar bucket), confirme que desea vaciar el bucket; para ello, ingrese **`permanently delete`** en el campo de texto y luego, elija Empty (Vaciar).

Eliminación del bucket de S3

Para eliminar los buckets de S3 que creó en Prerequisites (Requisitos previos), Paso 1 y Paso 5, realice lo siguiente:

1. Inicie sesión AWS Management Console Management Console y abra la consola de Amazon S3 en https://console.aws.amazon.com/s3/.
2. En el panel de navegación izquierdo, elija Instancias.
3. En la lista Buckets, elija el botón de opción junto al nombre del bucket que desea eliminar.
4. Elija Eliminar.
5. En la página Delete bucket (Eliminar bucket) confirme que desea eliminar el bucket introduciendo el nombre del bucket en el campo de texto y, a continuación, elija Delete bucket (Eliminar bucket).

Siguientes pasos

Después de completar este tutorial, puede explorar más a fondo otros casos de uso relevantes:

- Puede utilizar Amazon CloudFront para transmitir los archivos multimedia transcodificados a los espectadores de todo el mundo. Para obtener más información, consulte Tutorial: Alojamiento de video en streaming bajo demanda con Amazon S3, Amazon CloudFront y Amazon Route 53.
- Puede transcodificar videos en el momento en que los carga en el bucket fuente de S3. Para ello, puede configurar un desencadenador de eventos de Amazon S3 que invoque automáticamente la

función de Lambda para transcodificar nuevos objetos en S3 con MediaConvert. Para obtener más información, consulte Tutorial: Uso de un desencadenador de Amazon S3 para invocar una función de Lambda en la Guía para desarrolladores de AWS Lambda.

Tutorial: configuración de un sitio web estático en Amazon S3

> **Important**
>
> Amazon S3 aplica ahora el cifrado del servidor con claves administradas por Amazon S3 (SSE-S3) como el nivel básico de cifrado para cada bucket de Amazon S3. Desde el 5 de enero de 2023, todas las cargas de objetos nuevos a Amazon S3 se cifran automáticamente sin costo adicional y sin afectar al rendimiento. El estado de cifrado automático para la configuración de cifrado predeterminada en el bucket de S3 y para cargas de objetos nuevos está disponible en registros de AWS CloudTrail, Inventario de S3, Lente de almacenamiento de S3, la consola de Amazon S3 y como encabezado de respuesta a la API de Amazon S3 adicional en AWS Command Line Interface y los SDK de AWS. Para obtener más información, consulte Preguntas frecuentes del cifrado predeterminado.

Puede configurar un bucket de Amazon S3 para que funcione como un sitio web. En este ejemplo se muestran los pasos para el alojamiento de un sitio web en Amazon S3.

> **Important**
>
> Para el siguiente tutorial hay que desactivar la opción Bloqueo de acceso público. Le recomendamos que deje activada la opción Bloqueo de acceso público. Si desea conservar las cuatro configuraciones de Bloqueo de acceso público activadas y alojar un sitio web estático, puede utilizar el control de acceso de origen (OAC) de Amazon CloudFront. Amazon CloudFront proporciona las capacidades necesarias para configurar un sitio web estático seguro. Los sitios web estáticos de Amazon S3 solo admiten puntos de conexión HTTP. Amazon CloudFront utiliza el almacenamiento duradero de Amazon S3 a la vez que proporciona encabezados de seguridad adicionales, como HTTPS. HTTPS agrega seguridad al cifrar una solicitud HTTP normal y proteger contra ataques cibernéticos comunes. Para obtener información, consulte Introducción a un sitio web estático seguro en la guía para desarrolladores de Amazon CloudFront.

Temas

Paso 1: crear un bucket

Las siguientes instrucciones proporcionan información general sobre cómo crear los buckets para el alojamiento de sitios web. Para obtener instrucciones detalladas paso a paso sobre la creación de un bucket, consulte Crear un bucket.

Para crear un bucket

1. Inicie sesión en la AWS Management Console y abra la consola de Amazon S3 en https://console.aws.amazon.com/s3/.
2. Elija Create bucket (Crear bucket).
3. Introduzca el Bucket name (Nombre del bucket) (por ejemplo: **`example.com`**).
4. Elija la región en la que desea crear el bucket.

   Elija una región que esté cercana geográficamente para minimizar la latencia y los costos, o para cumplir los requisitos normativos. La región que elija determina el punto de conexión de sitio web de Amazon S3. Para obtener más información, consulte Puntos de enlace de sitio web.
5. Para aceptar la configuración predeterminada y crear el bucket, elija Create (Crear).

Paso 2: habilitar el alojamiento de un sitio web estático

Después de crear un bucket, puede habilitar el alojamiento de sitios web estático para su bucket. Puede crear un nuevo bucket o utilizar un bucket existente.

Para habilitar el alojamiento estático de sitios web

1. Inicie sesión en la AWS Management Console y abra la consola de Amazon S3 en https://console.aws.amazon.com/s3/.
2. En la lista Buckets, seleccione el nombre del bucket para el que desea habilitar el alojamiento de sitios web estáticos.
3. Seleccione Properties (Propiedades).
4. Elija Static website hosting (Alojamiento de sitios web estáticos), elija Edit (Editar).
5. Elija Use this bucket to host a website (Usar este bucket para alojar un sitio web).
6. En Static website hosting (Alojamiento de sitios web estáticos), elija Enable (Habilitar).
7. En Index Document (Documento de índice), escriba el nombre de archivo del documento de índice, normalmente `index.html`.

   El nombre del documento de índice distingue entre mayúsculas y minúsculas y debe coincidir exactamente con el nombre del archivo del documento de índice HTML que tiene previsto cargar en el bucket de S3. Al configurar un bucket para el alojamiento de sitios web, debe especificar un documento de índice. Amazon S3 devuelve este documento de índice cuando se reciben solicitudes en el dominio raíz o en cualquiera de las subcarpetas. Para obtener más información, consulte Configurar un documento de índice.
8. Si desea proporcionar su propio documento de error personalizado para los errores de clase 4XX, escriba el nombre de archivo del documento de error personalizado en Error document (Documento de error).

   El nombre del documento de error distingue entre mayúsculas y minúsculas y debe coincidir exactamente con el nombre del archivo del documento de error HTML que tiene previsto cargar en el bucket de S3. Si no especifica un documento de error personalizado y se produce un error, Amazon S3 devuelve un documento de error HTML predeterminado. Para obtener más información, consulte Configurar un documento de error personalizado.
9. (Opcional) Si desea especificar reglas de redireccionamiento avanzadas, en Redirection rules (Reglas de redireccionamiento), especifique JSON para describir las reglas.

   Por ejemplo, puede dirigir condicionalmente las solicitudes según nombres de clave de objeto o prefijos específicos en la solicitud. Para obtener más información, consulte Configurar reglas de redireccionamiento para utilizar redireccionamiento condicional avanzado.
10. Elija Save changes (Guardar cambios).

Amazon S3 permite activar el alojamiento de sitios web estáticos para su bucket. En la parte inferior de la página, en Static website hosting (Alojamiento de sitios web estáticos), verá el punto de conexión del sitio web para su bucket.

11. En Static website hosting (Alojamiento de sitios web estáticos), anote el valor de Endpoint (Punto de enlace).

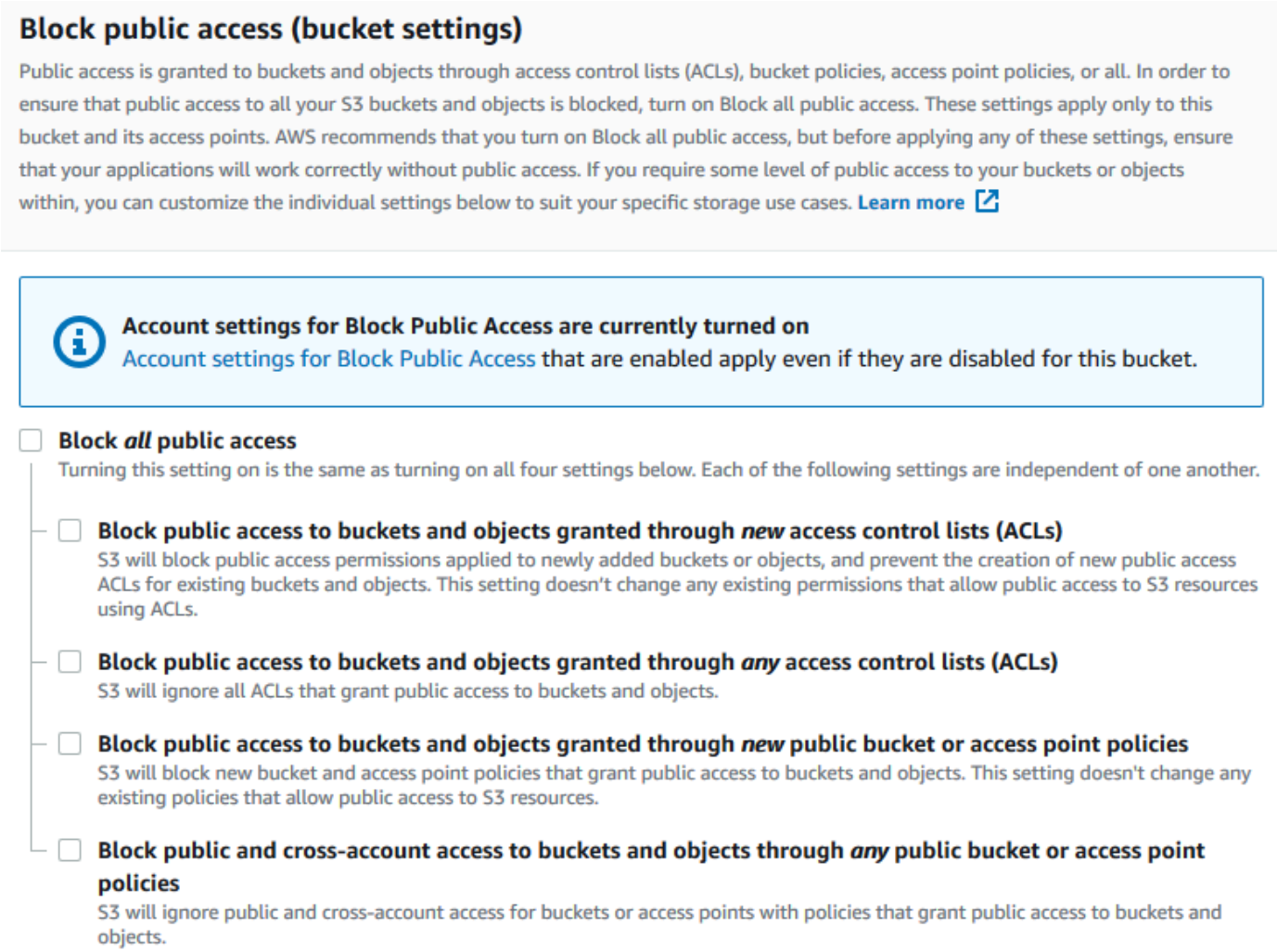
    Endpoint (Punto de enlace) es el punto de conexión del sitio web de Amazon S3 para el bucket. Cuando termine de configurar el bucket como un sitio web estático, puede utilizar este punto de conexión para probar el sitio web.

Paso 3: editar la configuración de bloqueo de acceso público

De forma predeterminada, Amazon S3 bloquea el acceso público a su cuenta y sus buckets. Si desea utilizar un bucket para alojar un sitio web estático, puede utilizar estos pasos para editar la configuración de bloqueo de acceso público.

> **Warning**
>
> Antes de completar estos pasos, consulte Bloquear el acceso público a su almacenamiento de Amazon S3 para asegurarse de que comprende y acepta los riesgos que implica permitir el acceso público. Cuando desactiva la configuración de acceso público de bloqueo para que el bucket sea público, cualquier usuario de Internet puede acceder al bucket. Le recomendamos que bloquee todo el acceso público a sus buckets.

1. Abra la consola de Amazon S3 en https://console.aws.amazon.com/s3.
2. Elija el nombre del bucket que ha configurado como sitio web estático.
3. Elija Permissions (Permisos).
4. En Block public access (bucket settings) (Bloquear acceso público [configuración de bucket]), elija Edit (Editar).
5. Desactive Block all public access (Bloquear todo el acceso público) y elija Save changes (Guardar cambios).



Amazon S3 desactiva la configuración de Bloqueo de acceso público para su bucket. Para crear un sitio web público y estático, es posible que también tenga que editar la configuración de Bloqueo de acceso público para su cuenta antes de agregar una política de bucket. Si la configuración de Bloqueo de acceso público de su cuenta está activada actualmente, verá una nota en Bloquear acceso público (configuración del bucket).

Paso 4: agregar una política de bucket para que el contenido del bucket sea público

Después de editar la configuración de acceso público de bloques de S3, debe agregar una política de bucket para garantizar el acceso de lectura público a su bucket. Cuando concede permiso de lectura público, cualquier persona de Internet puede acceder a su bucket.

> **Important**
>
> La política que se muestra a continuación es solo un ejemplo y permite acceso completo al contenido del bucket. Antes de continuar con este paso, revise ¿Cómo puedo proteger los archivos en mi bucket de Amazon S3? para asegurarse de que comprende las prácticas recomendadas para proteger los archivos en el bucket de S3 y los riesgos que implica la concesión de acceso público.

1. En Buckets, elija el nombre del bucket.
2. Elija Permissions (Permisos).
3. En Bucket Policy (Política de bucket), elija Edit (Editar).
4. Para conceder acceso público de lectura a su sitio web, copie la siguiente política de bucket y péguela en el Bucket policy editor (Editor de políticas de bucket).

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Sid": "PublicReadGetObject",
            "Effect": "Allow",
            "Principal": "*",
            "Action": [
                "s3:GetObject"
            ],
            "Resource": [
                "arn:aws:s3:::Bucket-Name/*"
            ]
        }
    ]
}
```

5. Actualice el valor de `Resource` para el nombre de su bucket.

   En la política de bucket de ejemplo anterior, *`Bucket-Name`* es un marcador de posición para el nombre del bucket. Para utilizar esta política de bucket con su propio bucket, debe actualizar este nombre para que coincida con su nombre de bucket.

6. Elija Guardar cambios.

Aparecerá un mensaje que indicará que la política de bucket se ha agregado correctamente.

Si ve un error que indica `Policy has invalid resource`, confirme que el nombre del bucket en la política del bucket coincide con el nombre de su bucket. Para obtener información acerca de cómo agregar una política de bucket, consulte ¿Cómo añado una política de bucket de S3?

Si recibe un mensaje de error y no puede guardar la política de bucket, compruebe la configuración del bloqueo de acceso público para la cuenta y el bucket para confirmar que permite acceso público al bucket.

Paso 5: configurar un documento de índice

Cuando habilite el alojamiento de sitio web estático para su bucket, escriba el nombre del documento de índice (por ejemplo, **`index.html`**). Después de habilitar el alojamiento de sitio web estático para el bucket, cargue un archivo HTML con el nombre de este documento de índice en el bucket.

Para configurar el documento de índice

1. Cree un archivo `index.html`.

   Si no tiene un archivo `index.html`, puede usar el siguiente HTML para crear uno:

   ```
   <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. Guarde el archivo de índice localmente.

   El nombre del archivo de documento de índice debe coincidir exactamente con el nombre del documento de índice que especifique en el cuadro de diálogo Static website hosting (Alojamiento de sitio web estático). El nombre del documento de índice distingue entre mayúsculas y minúsculas. Por ejemplo, si escribe `index.html` en el nombre del Index document (Documento

de índice) en el cuadro de diálogo Static website hosting (Alojamiento de sitio web estático), el nombre del archivo de documento de índice también debe ser `index.html` y no `Index.html`.

3. Inicie sesión en la AWS Management Console y abra la consola de Amazon S3 en https://console.aws.amazon.com/s3/.
4. En la lista Buckets, elija el nombre del bucket que desea utilizar para alojar un sitio web estático.
5. Habilite el alojamiento de sitios web estáticos para su bucket e introduzca el nombre exacto del documento de índice (por ejemplo: `index.html`). Para obtener más información, consulte Habilitar el alojamiento de sitios web.

   Después de habilitar el alojamiento estático del sitio web, continúe con el paso 6.
6. Para cargar el documento de índice en el bucket, realice una de las siguientes acciones:

   - Arrastre y suelte el archivo de índice en la lista de buckets de la consola.
   - Elija Upload (Cargar) y siga las instrucciones para elegir y cargar el archivo de índice.

   Para obtener instrucciones paso a paso, consulte Carga de objetos.
7. (Opcional) Cargue otros contenidos del sitio web en su bucket.

Paso 6: configurar un documento de error

Cuando habilite el alojamiento de sitios webs estáticos para el bucket, escriba el nombre del documento de error (por ejemplo, **`404.html`**). Después de habilitar el alojamiento de sitios web estáticos para el bucket, cargue un archivo HTML con el nombre de este documento de error en el bucket.

Para configurar un documento de error,

1. Cree un documento de error, por ejemplo `404.html`.
2. Guarde el archivo de documento de error localmente.

   El nombre del documento de error distingue mayúsculas y minúsculas y debe coincidir exactamente con el nombre que escriba al habilitar el alojamiento de sitios web estáticos. Por ejemplo, si escribe `404.html` en el nombre del Error document (Documento de error) en el cuadro de diálogo Static website hosting (Alojamiento de sitio web estático), el nombre del archivo del documento de error también debe ser `404.html`.

3. Inicie sesión en la AWS Management Console y abra la consola de Amazon S3 en https://console.aws.amazon.com/s3/.
4. En la lista Buckets, elija el nombre del bucket que desea utilizar para alojar un sitio web estático.
5. Habilite el alojamiento de sitios web estáticos para su bucket y escriba el nombre exacto del documento de error (por ejemplo: `404.html`). Para obtener más información, consulte Habilitar el alojamiento de sitios web y Configurar un documento de error personalizado.

   Después de habilitar el alojamiento estático del sitio web, continúe con el paso 6.
6. Para cargar el documento de error en el bucket, realice una de las siguientes acciones:

   - Arrastre y suelte el archivo del documento de error a la lista de buckets de la consola.
   - Elija Upload (Cargar) y siga las instrucciones para elegir y cargar el archivo de índice.

   Para obtener instrucciones paso a paso, consulte Carga de objetos.

Paso 7: probar el punto de conexión del sitio web

Después de configurar el alojamiento de sitios web estáticos para el bucket, puede probar el punto de conexión del sitio web.

> **Note**
>
> Amazon S3 no admite el acceso HTTPS al sitio web. Si desea usar HTTPS, puede emplear Amazon CloudFront para atender a un sitio web estático alojado en Amazon S3.
> Para obtener más información, consulte Cómo usar CloudFront para dar servicio a un sitio web estático alojado en Amazon S3 y Requerir HTTPS para la comunicación entre lectores y CloudFront.

1. En Buckets, elija el nombre del bucket.
2. Seleccione Properties (Propiedades).
3. En la parte inferior de la página, en Static website hosting (Alojamiento de sitios web estáticos), elija el punto de conexión del sitio web del bucket.

   El documento de índice se abre en una ventana independiente del explorador.

Ahora puede alojar un sitio web en Amazon S3. Este sitio web está disponible en el punto de conexión del sitio web de Amazon S3. Sin embargo, es posible que tenga un dominio, como `example.com`, que desee utilizar para distribuir el contenido del sitio web creado. Es posible que también desee utilizar el soporte para dominio raíz de Amazon S3 para distribuir las solicitudes para `http://www.example.com` y `http://example.com`. Esto requiere pasos adicionales. Para ver un ejemplo, consulte Tutorial: Configuración de un sitio web estático mediante un dominio personalizado registrado con Route 53.

Paso 8: eliminar

Si creó un sitio web estático solo como parte de un ejercicio de aprendizaje, elimine los recursos de AWS que asignó para dejar de acumular cargos. Después de que haya eliminado los recursos de AWS, el sitio web ya no estará disponible. Para obtener más información, consulte Eliminar un bucket.

Tutorial: Configuración de un sitio web estático mediante un dominio personalizado registrado con Route 53

Suponga que desea alojar un sitio web estático en Amazon S3. Ha registrado un dominio con Amazon Route 53 (por ejemplo, `example.com`) y desea que las solicitudes de contenido de Amazon S3 `http://www.example.com` y `http://example.com` se envíen desde él. Puede utilizar esta explicación para aprender a alojar un sitio web estático y crear redirecciones en Amazon S3 para un sitio web con un nombre de dominio personalizado registrado en Route 53. Puede trabajar con un sitio web existente que desee alojar en Amazon S3, o bien usar esta explicación para comenzar desde cero.

Una vez completada esta explicación, puede utilizar opcionalmente Amazon CloudFront para mejorar el rendimiento de su sitio web. Para obtener más información, consulte Aceleración de su sitio web con Amazon CloudFront.

> **Note**
>
> Los puntos de enlace del sitio web de Amazon S3 no admiten HTTPS ni puntos de acceso. Si desea usar HTTPS, puede emplear Amazon CloudFront para atender a un sitio web estático alojado en Amazon S3.
>
> Para ver un tutorial sobre cómo alojar su contenido de forma segura con CloudFront y Amazon S3, consulte Tutorial: Alojamiento de video en streaming bajo demanda con Amazon

S3, Amazon CloudFront y Amazon Route 53. Para obtener más información, consulte Cómo usar CloudFront para dar servicio a un sitio web estático alojado en Amazon S3 y Requerir HTTPS para la comunicación entre lectores y CloudFront.

Automatización de la configuración del sitio web estático con una plantilla de AWS CloudFormation

Puede utilizar una plantilla de AWS CloudFormation para automatizar la configuración de su sitio web estático. La plantilla de AWS CloudFormation configura los componentes que tiene que alojar en un sitio web estático seguro para que pueda centrarse más en el contenido de su sitio web y menos en la configuración de componentes.

La plantilla de AWS CloudFormation incluye los siguientes componentes:

- Amazon S3: crea un bucket de Amazon S3 para alojar su sitio web estático.
- CloudFront: crea una distribución de CloudFront para acelerar su sitio web estático.
- Lambda@Edge: usa Lambda@Edge para agregar encabezados de seguridad a cada respuesta del servidor. Los encabezados de seguridad son un grupo de encabezados en la respuesta del servidor web que indican a los navegadores web que tomen precauciones de seguridad adicionales. Para obtener más información, consulte esta entrada de blog: Adding HTTP security headers using Lambda@Edge and Amazon CloudFront.

Esta plantilla de AWS CloudFormation está disponible para que pueda descargarla y utilizarla. Para obtener información e instrucciones, consulte Introducción a un sitio web estático seguro en la guía para desarrolladores de Amazon CloudFront.

Temas

- Paso 8: editar la configuración del S3 Block Public Access
- Paso 9: adjuntar una política de bucket
- Paso 10: probar el punto de conexión del dominio
- Paso 11: agregar registros de alias para su dominio y subdominio
- Paso 12: probar el sitio web
- Aceleración de su sitio web con Amazon CloudFront
- Limpiar los recursos de ejemplo

Antes de empezar

A medida que siga los pasos de este ejemplo, trabajará con los siguientes servicios:

Amazon Route 53: puede utilizar Route 53 para registrar dominios y para definir a dónde quiere dirigir el tráfico de Internet para su dominio. El ejemplo muestra cómo crear registros de alias de Route 53 que dirigen el tráfico para su dominio (`example.com`) y subdominio (`www.example.com`) a un bucket de Amazon S3 que contiene un archivo HTML.

Amazon S3: puede utilizar Amazon S3 para crear buckets, cargar una página de sitio web de muestra, configurar permisos para que todos puedan ver el contenido y configurar los buckets para el alojamiento en el sitio web.

Paso 1: registrar un dominio personalizado con Route 53

Si aún no tiene un nombre de dominio registrado, como `example.com`, registre uno con Route 53. Para obtener más información, consulte Renovación de un nuevo dominio en la Guía para desarrolladores de Amazon Route 53. Después de registrar su nombre de dominio, puede crear y configurar sus buckets de Amazon S3 para el alojamiento de sitios web.

Paso 2: crear dos buckets

Para admitir solicitudes desde el dominio raíz y del subdominio, debe crear dos buckets:

- Bucket de dominio – `example.com`
- Bucket de subdominio – `www.example.com`

Estos nombres de bucket deben coincidir exactamente con su nombre de dominio. En este ejemplo, el nombre de dominio es `example.com`. Alojará su contenido fuera del bucket del dominio

raíz (`example.com`). Creará una solicitud de redireccionamiento para el bucket de subdominio (`www.example.com`). Si alguien escribe `www.example.com` en su navegador, se redirigen a `example.com` y ven el contenido que está alojado en el bucket de Amazon S3 con ese nombre.

Para crear los buckets para el alojamiento de sitios web

Las siguientes instrucciones proporcionan información general sobre cómo crear los buckets para el alojamiento de sitios web. Para obtener instrucciones detalladas paso a paso sobre la creación de un bucket, consulte Crear un bucket.

1. Inicie sesión en la AWS Management Console y abra la consola de Amazon S3 en https://console.aws.amazon.com/s3/.
2. Cree el bucket del dominio raíz:
   a. Elija Create bucket (Crear bucket).
   b. Introduzca el Bucket name (Nombre del bucket) (por ejemplo: **`example.com`**).
   c. Elija la región en la que desea crear el bucket.

      Elija una región que esté cercana geográficamente para minimizar la latencia y los costos, o para cumplir los requisitos normativos. La región que elija determina el punto de conexión de sitio web de Amazon S3. Para obtener más información, consulte Puntos de enlace de sitio web.
   d. Para aceptar la configuración predeterminada y crear el bucket, elija Create (Crear).
3. Cree el bucket del subdominio:
   a. Elija Create bucket (Crear bucket).
   b. Introduzca el Bucket name (Nombre del bucket) (por ejemplo: **`www.example.com`**).
   c. Elija la región en la que desea crear el bucket.

      Elija una región que esté cercana geográficamente para minimizar la latencia y los costos, o para cumplir los requisitos normativos. La región que elija determina el punto de conexión de sitio web de Amazon S3. Para obtener más información, consulte Puntos de enlace de sitio web.
   d. Para aceptar la configuración predeterminada y crear el bucket, elija Create (Crear).

En el paso siguiente, configure `example.com` para el alojamiento de sitio web.

Paso 3: configurar el bucket de dominio raíz para el alojamiento de sitios web

En este paso, configurará el bucket de dominio raíz (`example.com`) como un sitio web. Este bucket incluirá el contenido de su sitio web. Al configurar un bucket para el alojamiento de sitios web, puede acceder al sitio web a través de Puntos de enlace de sitio web.

Para habilitar el alojamiento estático de sitios web

1. Inicie sesión en la AWS Management Console y abra la consola de Amazon S3 en https://console.aws.amazon.com/s3/.
2. En la lista Buckets, seleccione el nombre del bucket para el que desea habilitar el alojamiento de sitios web estáticos.
3. Seleccione Properties (Propiedades).
4. Elija Static website hosting (Alojamiento de sitios web estáticos), elija Edit (Editar).
5. Elija Use this bucket to host a website (Usar este bucket para alojar un sitio web).
6. En Static website hosting (Alojamiento de sitios web estáticos), elija Enable (Habilitar).
7. En Index Document (Documento de índice), escriba el nombre de archivo del documento de índice, normalmente `index.html`.

   El nombre del documento de índice distingue entre mayúsculas y minúsculas y debe coincidir exactamente con el nombre del archivo del documento de índice HTML que tiene previsto cargar en el bucket de S3. Al configurar un bucket para el alojamiento de sitios web, debe especificar un documento de índice. Amazon S3 devuelve este documento de índice cuando se reciben solicitudes en el dominio raíz o en cualquiera de las subcarpetas. Para obtener más información, consulte Configurar un documento de índice.
8. Si desea proporcionar su propio documento de error personalizado para los errores de clase 4XX, escriba el nombre de archivo del documento de error personalizado en Error document (Documento de error).

   El nombre del documento de error distingue entre mayúsculas y minúsculas y debe coincidir exactamente con el nombre del archivo del documento de error HTML que tiene previsto cargar en el bucket de S3. Si no especifica un documento de error personalizado y se produce un error, Amazon S3 devuelve un documento de error HTML predeterminado. Para obtener más información, consulte Configurar un documento de error personalizado.

9. (Opcional) Si desea especificar reglas de redireccionamiento avanzadas, en Redirection rules (Reglas de redireccionamiento), especifique JSON para describir las reglas.

   Por ejemplo, puede dirigir condicionalmente las solicitudes según nombres de clave de objeto o prefijos específicos en la solicitud. Para obtener más información, consulte Configurar reglas de redireccionamiento para utilizar redireccionamiento condicional avanzado.
10. Elija Save changes (Guardar cambios).

    Amazon S3 permite activar el alojamiento de sitios web estáticos para su bucket. En la parte inferior de la página, en Static website hosting (Alojamiento de sitios web estáticos), verá el punto de conexión del sitio web para su bucket.
11. En Static website hosting (Alojamiento de sitios web estáticos), anote el valor de Endpoint (Punto de enlace).

    Endpoint (Punto de enlace) es el punto de conexión del sitio web de Amazon S3 para el bucket. Cuando termine de configurar el bucket como un sitio web estático, puede utilizar este punto de conexión para probar el sitio web.

Después de editar la configuración de bloqueo de acceso público y agregar una política de bucket que permita el acceso de lectura pública, puede utilizar el punto de conexión del sitio web para acceder a su sitio web.

En el paso siguiente, configure su subdominio (`www.example.com`) para redirigir las solicitudes a su dominio (`example.com`).

Paso 4: configurar el bucket de subdominio para el redireccionamiento del sitio web

Una vez que ha configurado su bucket de dominio raíz para el alojamiento de sitio web, puede configurar el bucket de subdominio para redireccionar todas las solicitudes al dominio. En este ejemplo, todas las solicitudes para `www.example.com` se redirigen a `example.com`.

Para configurar una solicitud de redirección, realice el siguiente procedimiento:

1. En la consola de Amazon S3, en la lista Buckets, elija su nombre de bucket de subdominio (`www.example.com`, en este ejemplo).
2. Seleccione Properties (Propiedades).
3. Elija Static website hosting (Alojamiento de sitios web estáticos), elija Edit (Editar).

4. Elija Redirect requests for an object (Redirigir solicitudes de un objeto).
5. En el cuadro Target bucket (Bucket de destino), escriba su dominio raíz (por ejemplo, **`example.com`**).
6. En Protocol (Protocolo), elija http.
7. Elija Save changes.

Paso 5: configurar registros para el tráfico del sitio web

Si desea hacer un seguimiento de la cantidad de visitas que acceden a su sitio web, puede habilitar el registro en el bucket del dominio raíz. Para obtener más información, consulte Registro de solicitudes con registro de acceso al servidor. Si tiene previsto utilizar Amazon CloudFront para acelerar su sitio web, también puede usar el registro de CloudFront.

Para habilitar el registro de acceso al servidor para el bucket del dominio raíz

1. Abra la consola de Amazon S3 en https://console.aws.amazon.com/s3.
2. En la misma región en la que creó el bucket configurado como sitio web estático, cree un bucket para el registro, por ejemplo `logs.example.com`.
3. Cree una carpeta para los archivos de registro de acceso al servidor (por ejemplo: `logs`).
4. (Opcional) Si desea utilizar CloudFront para mejorar el rendimiento del sitio web, cree una carpeta para los archivos de registro de CloudFront (por ejemplo: `cdn`).

> **Important**
>
> Al crear o actualizar una distribución y habilitar el registro de CloudFront, CloudFront actualiza la lista de control de acceso (ACL) del bucket para conceder a la cuenta `awslogsdelivery` permisos `FULL_CONTROL` para que escriba registros en el bucket. Para obtener más información, consulte Permisos necesarios para configurar el registro estándar y acceder a los archivos de registro en la Guía para desarrolladores de Amazon CloudFront. Si el bucket que almacena los registros utiliza la configuración de propietario del bucket obligatorio de S3 Object Ownership para desactivar las ACL, CloudFront no puede escribir registros en el bucket. Para obtener más información, consulte Control de la propiedad de los objetos y desactivación de las ACL del bucket.

5. En la lista Buckets, elija el bucket de dominio raíz.
6. Seleccione Properties (Propiedades).

7. En Server access logging (Registro de acceso al servidor), elija Edit (Editar).
8. Elija Enable.
9. En el bucket de destino, elija el destino del bucket y la carpeta para los registros de acceso al servidor:

   - Busque la carpeta y la ubicación del bucket:
     1. Elija Browse S3 (Examinar S3).
     2. Elija el nombre del bucket y, a continuación, elija la carpeta de registros.
     3. Elija Choose path (Elegir ruta).
   - Introduzca la ruta del bucket de S3, por ejemplo, `s3://logs.example.com/logs/`.
10. Elija Save changes (Guardar cambios).

    En su bucket de registro, ahora puede acceder a sus registros. Amazon S3 escribe los registros de acceso al sitio web en su bucket de registro cada dos horas.

Paso 6: cargar índice y contenido del sitio web

En este paso, cargue el documento de índice y el contenido del sitio web opcional en el bucket de dominio raíz.

Cuando habilite el alojamiento de sitio web estático para su bucket, escriba el nombre del documento de índice (por ejemplo: **`index.html`**). Después de habilitar el alojamiento de sitio web estático para el bucket, cargue un archivo HTML con el nombre de este documento de índice en el bucket.

Para configurar el documento de índice

1. Cree un archivo `index.html`.

   Si no tiene un archivo `index.html`, puede usar el siguiente HTML para crear uno:

```
<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. Guarde el archivo de índice localmente.

   El nombre del archivo de documento de índice debe coincidir exactamente con el nombre del documento de índice que especifique en el cuadro de diálogo Static website hosting (Alojamiento de sitio web estático). El nombre del documento de índice distingue entre mayúsculas y minúsculas. Por ejemplo, si escribe `index.html` en el nombre del Index document (Documento de índice) en el cuadro de diálogo Static website hosting (Alojamiento de sitio web estático), el nombre del archivo de documento de índice también debe ser `index.html` y no `Index.html`.
3. Inicie sesión en la AWS Management Console y abra la consola de Amazon S3 en https://console.aws.amazon.com/s3/.
4. En la lista Buckets, elija el nombre del bucket que desea utilizar para alojar un sitio web estático.
5. Habilite el alojamiento de sitios web estáticos para su bucket e introduzca el nombre exacto del documento de índice (por ejemplo: `index.html`). Para obtener más información, consulte Habilitar el alojamiento de sitios web.

   Después de habilitar el alojamiento estático del sitio web, continúe con el paso 6.
6. Para cargar el documento de índice en el bucket, realice una de las siguientes acciones:

   - Arrastre y suelte el archivo de índice en la lista de buckets de la consola.
   - Elija Upload (Cargar) y siga las instrucciones para elegir y cargar el archivo de índice.

   Para obtener instrucciones paso a paso, consulte Carga de objetos.
7. (Opcional) Cargue otros contenidos del sitio web en su bucket.

Paso 7: cargar un documento de error

Cuando habilite el alojamiento de sitios webs estáticos para el bucket, escriba el nombre del documento de error (por ejemplo: **`404.html`**). Después de habilitar el alojamiento de sitios web estáticos para el bucket, cargue un archivo HTML con el nombre de este documento de error en el bucket.

Para configurar un documento de error,

1. Cree un documento de error, por ejemplo `404.html`.
2. Guarde el archivo de documento de error localmente.

El nombre del documento de error distingue mayúsculas y minúsculas y debe coincidir exactamente con el nombre que escriba al habilitar el alojamiento de sitios web estáticos. Por ejemplo, si escribe `404.html` en el nombre del Error document (Documento de error) en el cuadro de diálogo Static website hosting (Alojamiento de sitio web estático), el nombre del archivo del documento de error también debe ser `404.html`.

3. Inicie sesión en la AWS Management Console y abra la consola de Amazon S3 en https://console.aws.amazon.com/s3/.
4. En la lista Buckets, elija el nombre del bucket que desea utilizar para alojar un sitio web estático.
5. Habilite el alojamiento de sitios web estáticos para su bucket y escriba el nombre exacto del documento de error (por ejemplo: `404.html`). Para obtener más información, consulte Habilitar el alojamiento de sitios web y Configurar un documento de error personalizado.

   Después de habilitar el alojamiento estático del sitio web, continúe con el paso 6.
6. Para cargar el documento de error en el bucket, realice una de las siguientes acciones:

   - Arrastre y suelte el archivo del documento de error a la lista de buckets de la consola.
   - Elija Upload (Cargar) y siga las instrucciones para elegir y cargar el archivo de índice.

   Para obtener instrucciones paso a paso, consulte Carga de objetos.

Paso 8: editar la configuración del S3 Block Public Access

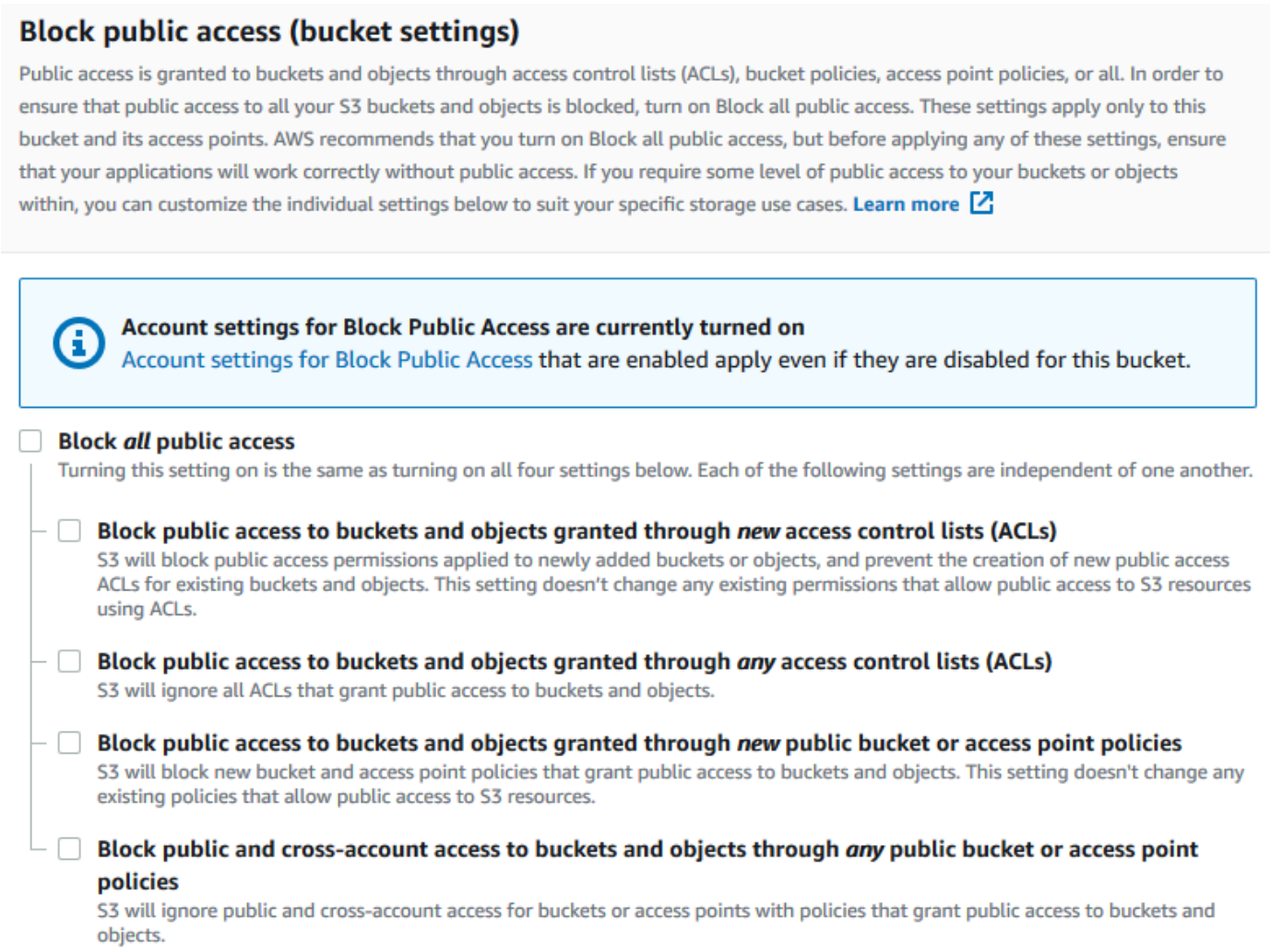
En este ejemplo, se edita la configuración de acceso público del bloque en el bucket del dominio (`example.com`) para permitir el acceso público.

De forma predeterminada, Amazon S3 bloquea el acceso público a su cuenta y sus buckets. Si desea utilizar un bucket para alojar un sitio web estático, puede utilizar estos pasos para editar la configuración de bloqueo de acceso público.

> **Warning**
>
> Antes de completar estos pasos, consulte Bloquear el acceso público a su almacenamiento de Amazon S3 para asegurarse de que comprende y acepta los riesgos que implica permitir el acceso público. Cuando desactiva la configuración de acceso público de bloqueo para

que el bucket sea público, cualquier usuario de Internet puede acceder al bucket. Le recomendamos que bloquee todo el acceso público a sus buckets.

1. Abra la consola de Amazon S3 en https://console.aws.amazon.com/s3.
2. Elija el nombre del bucket que ha configurado como sitio web estático.
3. Elija Permissions (Permisos).
4. En Block public access (bucket settings) (Bloquear acceso público [configuración de bucket]), elija Edit (Editar).
5. Desactive Block all public access (Bloquear todo el acceso público) y elija Save changes (Guardar cambios).



Amazon S3 desactiva la configuración de Bloqueo de acceso público para su bucket. Para crear un sitio web público y estático, es posible que también tenga que editar la configuración

de Bloqueo de acceso público para su cuenta antes de agregar una política de bucket. Si la configuración de Bloqueo de acceso público de su cuenta está activada actualmente, verá una nota en Bloquear acceso público (configuración del bucket).

Paso 9: adjuntar una política de bucket

En este ejemplo, se adjunta una política de bucket al bucket de dominio (`example.com`) para permitir el acceso de lectura pública. Reemplaza el *`Bucket-Name`* de la política de bucket de ejemplo por el nombre del bucket de dominio, por ejemplo `example.com`.

Después de editar la configuración de acceso público de bloques de S3, debe agregar una política de bucket para garantizar el acceso de lectura público a su bucket. Cuando concede permiso de lectura público, cualquier persona de Internet puede acceder a su bucket.

> **Important**
>
> La política que se muestra a continuación es solo un ejemplo y permite acceso completo al contenido del bucket. Antes de continuar con este paso, revise ¿Cómo puedo proteger los archivos en mi bucket de Amazon S3? para asegurarse de que comprende las prácticas recomendadas para proteger los archivos en el bucket de S3 y los riesgos que implica la concesión de acceso público.

1. En Buckets, elija el nombre del bucket.
2. Elija Permissions (Permisos).
3. En Bucket Policy (Política de bucket), elija Edit (Editar).
4. Para conceder acceso público de lectura a su sitio web, copie la siguiente política de bucket y péguela en el Bucket policy editor (Editor de políticas de bucket).

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Sid": "PublicReadGetObject",
            "Effect": "Allow",
            "Principal": "*",
            "Action": [
                "s3:GetObject"
```

```
            ],
            "Resource": [
                "arn:aws:s3:::Bucket-Name/*"
            ]
        }
    ]
}
```

5. Actualice el valor de `Resource` para el nombre de su bucket.

   En la política de bucket de ejemplo anterior, *`Bucket-Name`* es un marcador de posición para el nombre del bucket. Para utilizar esta política de bucket con su propio bucket, debe actualizar este nombre para que coincida con su nombre de bucket.

6. Elija Guardar cambios.

   Aparecerá un mensaje que indicará que la política de bucket se ha agregado correctamente.

   Si ve un error que indica `Policy has invalid resource`, confirme que el nombre del bucket en la política del bucket coincide con el nombre de su bucket. Para obtener información acerca de cómo agregar una política de bucket, consulte ¿Cómo añado una política de bucket de S3?

   Si recibe un mensaje de error y no puede guardar la política de bucket, compruebe la configuración del bloqueo de acceso público para la cuenta y el bucket para confirmar que permite acceso público al bucket.

En el paso siguiente, puede determinar los puntos de enlace de su sitio web y probar el punto de conexión de su dominio.

Paso 10: probar el punto de conexión del dominio

Después de configurar el bucket de dominio para alojar un sitio web público, puede probar el punto de conexión de su dominio. Para obtener más información, consulte Puntos de enlace de sitio web. Solo podrá probar el punto de conexión para su bucket de dominio, ya que este está configurado para el redireccionamiento de sitios web y no para el alojamiento de sitios web estáticos.

> **Note**
>
> Amazon S3 no admite el acceso HTTPS al sitio web. Si desea usar HTTPS, puede emplear Amazon CloudFront para atender a un sitio web estático alojado en Amazon S3.

Para obtener más información, consulte Cómo usar CloudFront para dar servicio a un sitio web estático alojado en Amazon S3 y Requerir HTTPS para la comunicación entre lectores y CloudFront.

1. En Buckets, elija el nombre del bucket.
2. Seleccione Properties (Propiedades).
3. En la parte inferior de la página, en Static website hosting (Alojamiento de sitios web estáticos), elija el punto de conexión del sitio web del bucket.

   El documento de índice se abre en una ventana independiente del explorador.

En el siguiente paso, utiliza Amazon Route 53 para permitir a los clientes utilizar sus URL personalizadas para navegar a su sitio.

Paso 11: agregar registros de alias para su dominio y subdominio

En este paso, se crean los registros de alias que añade a la zona alojada para sus mapeos de dominio `example.com` y `www.example.com`. En lugar de utilizar direcciones IP, los registros de alias utilizan puntos de enlace de sitio web de Amazon S3. Amazon Route 53 mantiene el mapeo entre los registros de alias y las direcciones IP donde residen los buckets de Amazon S3. Se crean dos registros de alias, uno para el dominio raíz y otro para el subdominio.

Agregar un registro de alias para el dominio raíz y el subdominio

Para agregar un registro de alias para su dominio raíz (**`example.com`**)

1. Abra la consola de Route 53 en https://console.aws.amazon.com/route53/.

   > **Note**
   >
   > Si aún no utiliza Route 53, consulte el Paso 1: registrar un dominio en la Guía para desarrolladores de Amazon Route 53. Después de completar la configuración, puede reanudar las instrucciones.

2. Elija Hosted zones (Zonas alojadas).
3. En la lista de zonas alojadas, elija el nombre de la zona alojada que coincide con su nombre de dominio.

4. Elija Create record (Crear registro).
5. Elija Switch to wizard (Cambiar al asistente).

> **Note**
>
> Si desea utilizar la creación rápida para crear sus registros de alias, consulte Configuring Route 53 to route traffic to an S3 Bucket (Configuración de Route 53 para dirigir el tráfico a un bucket de S3).

6. Elija Simple routing (Direccionamiento sencillo) y Next (Siguiente).
7. Elija Define simple record (Definir registro simple).
8. En Record name (Nombre de registro) acepte el valor predeterminado, que es el nombre de la zona alojada y el dominio.
9. En Value/Route traffic to (Valor/Dirigir tráfico a), elija Alias to S3 website endpoint (Alias a punto de conexión de sitio web de S3).
10. Elija la región.
11. Elija el bucket de S3.

    El nombre del bucket debe coincidir con el nombre que aparece en el cuadro Name (Nombre). En la lista Elegir bucket de S3 , el nombre del bucket aparece con el punto de conexión del sitio web de Amazon S3 para la región donde se creó el bucket, por ejemplo, `s3-website-us-west-1.amazonaws.com (example.com)`.

    Choose S3 bucket (Elegir bucket de S3) muestra un bucket si:

    - Configuró el bucket como un sitio web estático.
    - El nombre del bucket es el mismo que el del registro que está creando.
    - La Cuenta de AWS actual creó el bucket.

    Si el bucket no aparece en la descripción de Choose S3 bucket (Elegir bucket de S3), escriba el punto de conexión del sitio web de Amazon S3 de la región en la que se creó el bucket, por ejemplo, **`s3-website-us-west-2.amazonaws.com`**. Para obtener una lista completa de los puntos de enlace del sitio web de Amazon S3, consulte Puntos de enlace de sitio web de Amazon S3. Para obtener más información acerca del destino de alias, consulte Value/route traffic to (Valor/ruta de destino del tráfico) en la Guía para desarrolladores de Amazon Route 53.
12. En Tipo de registro, elija A: Dirige el tráfico a una dirección IPv4 y algunos recursos de AWS.

13. En Evaluate target health (Evaluar el estado del destino), elija No.
14. Elija Define simple record (Definir registro simple).

Para agregar un registro de alias para su subdominio (**`www.example.com`**)

1. En Configure records (Configurar registros), elija Define simple record (Definir registro simple).
2. En Record name (Nombre de registro) para el subdominio, escriba `www`.
3. En Valor/Dirigir tráfico a, elija Alias a punto de conexión de sitio web de S3.
4. Elija la región.
5. Elija el bucket de S3; por ejemplo, `s3-website-us-west-2.amazonaws.com (www.example.com)`.

   Si el bucket no aparece en la descripción de Elegir bucket de S3, escriba el punto de conexión del sitio web de Amazon S3 de la región en la que se creó el bucket, por ejemplo, **`s3-website-us-west-2.amazonaws.com`**. Para obtener una lista completa de los puntos de enlace del sitio web de Amazon S3, consulte Puntos de enlace de sitio web de Amazon S3. Para obtener más información acerca del destino de alias, consulte Value/route traffic to (Valor/ruta de destino del tráfico) en la Guía para desarrolladores de Amazon Route 53.
6. En Tipo de registro, elija A: Dirige el tráfico a una dirección IPv4 y algunos recursos de AWS.
7. En Evaluate target health (Evaluar el estado del destino), elija No.
8. Elija Define simple record (Definir registro simple).
9. En la página Configure records (Configurar registros) , elija Create records (Crear registros).

> **Note**
>
> Por lo general, los cambios se propagan a todos los servidores de Route 53 en un plazo de 60 segundos. Una vez finalizada la propagación, puede dirigir el tráfico a su bucket de Amazon S3 mediante los nombres de los registros de alias que ha creado en este procedimiento.

Adición de un registro de alias para el dominio raíz y el subdominio (consola de Route 53 antigua)

Para agregar un registro de alias para su dominio raíz (**`example.com`**)

Se ha rediseñado la consola de Route 53. En la consola de Route 53 puede usar temporalmente la consola anterior. Si decide trabajar con la consola de Route 53 antigua, siga el procedimiento que se muestra a continuación.

1. Abra la consola de Route 53 en https://console.aws.amazon.com/route53/.

   > **Note**
   >
   > Si aún no utiliza Route 53, consulte el Paso 1: registrar un dominio en la Guía para desarrolladores de Amazon Route 53. Después de completar la configuración, puede reanudar las instrucciones.

2. Elija Hosted Zones (Zonas alojadas).
3. En la lista de zonas alojadas, elija el nombre de la zona alojada que coincide con su nombre de dominio.
4. Elija Create Record Set (Crear conjunto de registros).
5. Especifique los valores siguientes:

   Nombre

   Acepte el valor predeterminado, que es el nombre de la zona alojada y el dominio.

   Para el dominio raíz, no tiene que introducir ninguna información adicional en el campo Name (Nombre).

   Tipo

   Elija A - IPv4 address (A - Dirección IPv4).

   Alias

   Seleccione Yes.

   Alias Target

   En la sección de Puntos de enlace del sitio web de S3 de la lista, elija el nombre del bucket.

El nombre del bucket debe coincidir con el nombre que aparece en el cuadro Name (Nombre). En la descripción de Destino del alias, el nombre del bucket va seguido del punto de conexión del sitio web de Amazon S3 de la región donde se creó el bucket, por ejemplo `example.com (s3-website-us-west-2.amazonaws.com)`. Alias Target (Destino de alias) muestra un bucket si:

- Configuró el bucket como un sitio web estático.
- El nombre del bucket es el mismo que el del registro que está creando.
- La Cuenta de AWS actual creó el bucket.

Si el bucket no aparece en la descripción de Destino del alias, escriba el punto de conexión del sitio web de Amazon S3 de la región en la que se creó el bucket, por ejemplo, `s3-website-us-west-2`. Para obtener una lista completa de los puntos de enlace del sitio web de Amazon S3, consulte Puntos de enlace de sitio web de Amazon S3. Para obtener más información acerca del destino de alias, consulte Value/route traffic to (Valor/ruta de destino del tráfico) en la Guía para desarrolladores de Amazon Route 53.

Routing Policy

Acepte el valor predeterminado de Simple.

Evaluate Target Health

Acepte el valor predeterminado de No.

6. Seleccione Create (Crear).

Para agregar un registro de alias para su subdominio (**`www.example.com`**)

1. En la zona alojada para su dominio raíz (`example.com`), elija Create Record Set (Crear conjunto de registros).
2. Especifique los valores siguientes:

   Nombre

   Para el subdominio, introduzca `www` en el recuadro.

   Tipo

   Elija A - IPv4 address (A - Dirección IPv4).

Alias

Seleccione Yes.

Alias Target

En la sección S3 website endpoints (Puntos de enlace de sitio web de S3) de la lista, elija el mismo nombre de bucket que aparece en el campo Name (Nombre), por ejemplo `www.example.com (s3-website-us-west-2.amazonaws.com)`.

Routing Policy

Acepte el valor predeterminado de Simple.

Evaluate Target Health

Acepte el valor predeterminado de No.

3. Seleccione Create (Crear).

> **Note**
>
> Por lo general, los cambios se propagan a todos los servidores de Route 53 en un plazo de 60 segundos. Una vez finalizada la propagación, puede dirigir el tráfico a su bucket de Amazon S3 mediante los nombres de los registros de alias que ha creado en este procedimiento.

Paso 12: probar el sitio web

Compruebe que el sitio web y el redireccionamiento funcionan correctamente. En el navegador, escriba sus URL. En este ejemplo, puede probar las siguientes URL:

- Domain (Dominio) (`http://example.com`): muestra el documento de índice en el bucket `example.com`.
- Subdomain (Subdominio) (`http://www.example.com`): redirige la solicitud a `http://example.com`. Verá el documento de índice en el bucket `example.com`.

Si el sitio web o los vínculos de redirección no funcionan, puede probar lo siguiente:

- Clear cache (Borrar caché): borra la caché de su navegador web.

- Check name servers (Comprobar servidores de nombres): si la página web y los vínculos de redireccionamiento no funcionan después de haber borrado la caché, puede comparar los servidores de nombres de su dominio y los servidores de nombres de su zona alojada. Si los servidores de nombres no coinciden, es posible que deba actualizar los servidores de nombres de dominio para que coincidan con los que figuran en la zona alojada. Para obtener más información, consulte Agregar o cambiar servidores de nombres y pegar registros para un dominio.

Una vez que haya probado correctamente el dominio raíz y el subdominio, puede configurar una distribución de Amazon CloudFront para mejorar el rendimiento de su sitio web y proporcionar registros que puede utilizar para revisar el tráfico del sitio web. Para obtener más información, consulte Aceleración de su sitio web con Amazon CloudFront.

Aceleración de su sitio web con Amazon CloudFront

Puede usar Amazon CloudFront para mejorar el rendimiento del sitio web de Amazon S3. CloudFront pone los archivos de su sitio web (archivos HTML, imágenes y vídeos) a disposición desde los centros de datos de todo el mundo. Estos centros de datos se conocen como ubicaciones de borde. Cuando un visitante solicita un archivo de su sitio web, CloudFront redirecciona automáticamente la solicitud a una copia del archivo en la ubicación de borde más cercana. Esto genera tiempos de descarga menores que los que se obtendrían si el visitante solicitara el contenido desde un centro de datos más lejano.

CloudFront copia en caché el contenido en las ubicaciones de borde durante el período que usted especifique. Si un visitante solicita contenido que se ha copiado en caché y ha excedido la fecha de vencimiento, CloudFront accede al servidor de origen para verificar si hay disponible una versión más reciente del contenido. Si se encuentra una versión más reciente, CloudFront copiará esta nueva versión en la ubicación de borde. Los cambios que realice en el contenido original se replicarán en las ubicaciones de borde a medida que los visitantes soliciten el contenido.

Uso de CloudFront sin Route 53

En los tutoriales de esta página, se utiliza Route 53 para apuntar a la distribución de CloudFront. Sin embargo, si quiere ofrecer contenido alojado en un bucket de Amazon S3 mediante CloudFront sin utilizar Route 53, consulte Amazon CloudFront Tutorials: Setting up a Dynamic Content Distribution for Amazon S3 (Tutoriales de Amazon CloudFront: Configuración de una distribución dinámica de contenido para Amazon S3). Al ofrecer contenido alojado en un bucket de Amazon S3 mediante CloudFront, puede utilizar cualquier nombre de bucket y se admite tanto HTTP como HTTPS.

Automatizar la configuración con una plantilla de AWS CloudFormation

Con el fin de obtener más información acerca del uso de una plantilla de AWS CloudFormation para configurar un sitio web estático seguro que cree una distribución de CloudFront para dar servicio al sitio web, consulte Introducción a un sitio web estático seguro en la Guía para desarrolladores de Amazon CloudFront.

Temas

Paso 1: crear una distribución de CloudFront

Primero, cree una distribución de CloudFront. Esto habilita el acceso a su sitio web desde los centros de datos de todo el mundo.

Para crear una distribución con un origen de Amazon S3, realice las siguientes acciones:

1. Abra la consola de CloudFront en https://console.aws.amazon.com/cloudfront/v4/home.
2. Elija Crear distribución.
3. En la página Crear distribución, sección Configuración del origen, para Nombre de dominio de origen, escriba el punto de conexión del sitio web de Amazon S3 para su bucket, por ejemplo, **`example.com.s3-website.us-west-1.amazonaws.com`**.

   CloudFront completa el campo Origin ID (ID de origen) por usted.
4. En Configuración predeterminada de comportamiento de caché, conserve los valores predeterminados.

   Con la configuración predeterminada de Viewer Protocol Policy (Política del protocolo del lector), puede usar HTTPS para el sitio web estático. Para obtener más información de estas opciones de configuración, consulte Valores que deben especificarse al crear o actualizar una distribución web en la guía para desarrolladores de Amazon CloudFront.
5. En Distribution Settings (Configuración de distribución), haga lo siguiente:

   a. En Price Class (Clase de precio), deje la opción Use All Edge Locations (Best Performance) (Usar todas las ubicaciones de borde [mejor rendimiento]).

b. Establezca Alternate Domain Names (CNAMEs) (Nombres del dominio alternativos [CNAME]) en el dominio raíz y el subdominio `www`. En este tutorial, estos son `example.com` y `www.example.com`.

> **Important**
>
> Antes de realizar este paso, tenga en cuenta los requisitos para el uso de nombres de dominio alternativos, en concreto la necesidad de tener un certificado SSL/TLS válido.

c. En SSL Certificate (Certificado SSL), elija Custom SSL Certificate (example.com) (Certificado SSL personalizado [ejemplo.com]) y elija el certificado personalizado que cubra los nombres de dominio y subdominio.

   Para obtener más información, consulte Certificado SSL en la guía para desarrolladores de Amazon CloudFront.

d. En Objeto raíz predeterminado, introduzca el nombre del documento de índice, por ejemplo, `index.html`.

   Si la dirección URL utilizada para acceder a la distribución no contiene un nombre de archivo, la distribución de CloudFront devuelve el documento de índice. El objeto raíz predeterminado debe coincidir exactamente con el nombre del documento de índice de su sitio web estático. Para obtener más información, consulte Configurar un documento de índice.

e. En Logging (Registro), seleccione la opción On (Activado).

> **Important**
>
> Al crear o actualizar una distribución y habilitar el registro de CloudFront, CloudFront actualiza la lista de control de acceso (ACL) del bucket para conceder a la cuenta `awslogsdelivery` permisos `FULL_CONTROL` para que escriba registros en el bucket. Para obtener más información, consulte Permisos necesarios para configurar el registro estándar y acceder a los archivos de registro en la Guía para desarrolladores de Amazon CloudFront. Si el bucket que almacena los registros utiliza la configuración de propietario del bucket obligatorio de S3 Object Ownership para desactivar las ACL, CloudFront no puede escribir registros en el bucket.

Para obtener más información, consulte Control de la propiedad de los objetos y desactivación de las ACL del bucket.

f. En Bucket for Logs (Bucket para registros), seleccione el bucket de registro que creó.

   Para obtener más información acerca de la configuración de un bucket de registro, consulte (Opcional) Registro del tráfico web.

g. Si desea almacenar los registros generados por el tráfico en la distribución en una carpeta de CloudFront, en Log Prefix (Prefijo de registro), escriba el nombre de la carpeta.

h. No cambie ningún otro valor predeterminado.

6. Seleccione Create Distribution (Crear distribución).
7. Para ver el estado de la distribución, busque la distribución en la consola y revise la columna Status (Estado).

   El estado `InProgress` indica que la implementación de la distribución no ha finalizado aún.

   Una vez que haya implementado la distribución, puede hacer referencia al contenido con el nuevo nombre del dominio de CloudFront.
8. Registre el valor de Domain Name (Nombre de dominio) que se muestra en la consola de CloudFront, por ejemplo, `dj4p1rv6mvubz.cloudfront.net`.
9. Para verificar que la distribución de CloudFront funcione correctamente, escriba el nombre del dominio de la distribución en el navegador web.

   Si su sitio web es visible, la distribución de CloudFront funciona. Si su sitio web tiene un dominio personalizado registrado con Amazon Route 53, necesitará el nombre de dominio de CloudFront para actualizar el registro establecido en el siguiente paso.

Paso 2: Actualizar los conjuntos de registros para su dominio y subdominio

Como ya ha creado con éxito una distribución de CloudFront, actualice el registro de alias en Route 53 para que se asocien a la nueva distribución de CloudFront.

Para actualizar el registro de alias para que apunte a una distribución de CloudFront, realice las siguientes tareas:

1. Abra la consola de Route 53 en https://console.aws.amazon.com/route53/.
2. En el panel de navegación izquierdo, elija Hosted zones (Zonas alojadas).

3. En la página Hosted zones (Zonas alojadas), elija la zona alojada que creó para su subdominio, como, por ejemplo, `www.example.com`.
4. En Records (Registros), seleccione el registro A que creó para el subdominio.
5. En Record details (Detalles del registro), elija Edit record (Editar registro).
6. En Route traffic to (Dirigir tráfico a), elija Alias to Cloudfront distribution (Alias a distribución de CloudFront).
7. En Choose distribution (Elegir distribución), elija la distribución de CloudFront.
8. Seleccione Save.
9. Para redirigir el registro A del dominio raíz a la distribución de CloudFront, repita este procedimiento del dominio raíz, por ejemplo, `example.com`.

   La actualización de los conjuntos de registros se realiza entre 2 y 48 horas.
10. Para ver si los nuevos registros A han entrado en vigor, en un navegador web, introduzca la URL de su subdominio, por ejemplo, `http://www.example.com`.

    Si el navegador ya no lo redirige al dominio raíz (por ejemplo, `http://example.com`), los nuevos registros A están en su lugar. Cuando se haya aplicado el nuevo registro A, el tráfico redirigido por el nuevo registro A a la distribución de CloudFront no se redirige al dominio raíz. Todos los visitantes que hagan referencia al sitio mediante `http://example.com` o `http://www.example.com` se redirigirán a la ubicación de borde de CloudFront más cercana. De esta manera, los tiempos de descarga serán más rápidos.

> **Tip**
>
> Los navegadores pueden copiar en caché los ajustes de redirección. Si cree que se deberían haber aplicado los ajustes del nuevo registro A, pero el navegador aún redirige `http://www.example.com` a `http://example.com`, intente borrar el historial de navegador y las copias en caché y, luego, cierre y vuelva a abrir el navegador. También puede intentar con otro navegador web.

(Opcional) Paso 3: comprobar los archivos de registro

Los registros de acceso le informan cuántas personas visitan el sitio web. También contienen datos comerciales valiosos que puede analizar con otros servicios, como Amazon EMR.

Los registros de CloudFront se almacenan en el bucket y la carpeta que elija al crear una distribución de CloudFront y habilitar el registro. CloudFront escribe registros en su bucket de registros dentro de las 24 horas en las que se realizaron las solicitudes correspondientes.

Para ver los archivos de registro de su sitio web

1. Abra la consola de Amazon S3 en https://console.aws.amazon.com/s3.
2. Seleccione el nombre del bucket de registro para su sitio web.
3. Elija la carpeta de registros de CloudFront.
4. Descargue los archivos `.gzip` escritos por CloudFront antes de abrirlos.

   Si creó un sitio web solo como parte de un ejercicio de aprendizaje, puede eliminar los recursos que asignó para dejar de acumular cargos. Para ello, consulte Limpiar los recursos de ejemplo. Después de que haya eliminado los recursos de AWS, el sitio web ya no estará disponible.

Limpiar los recursos de ejemplo

Si creó un sitio web estático como parte de un ejercicio de aprendizaje, debe eliminar los recursos de AWS que asignó para dejar de acumular cargos. Después de que haya eliminado los recursos de AWS, el sitio web ya no estará disponible.

Tareas

- Paso 1: eliminar la distribución de Amazon CloudFront
- Paso 2: eliminar la zona alojada en Route 53
- Paso 3: deshabilitar el registro y eliminar el bucket de S3

Paso 1: eliminar la distribución de Amazon CloudFront

Antes de eliminar una distribución de Amazon CloudFront, debe desactivarla. Una distribución deshabilitada ya no es funcional y no acumula cargos. Puede habilitar una distribución deshabilitada en cualquier momento. Después de eliminar una distribución deshabilitada, ya no estará disponible.

Para desactivar y eliminar una distribución de CloudFront, realice las siguientes acciones:

1. Abra la consola de CloudFront en https://console.aws.amazon.com/cloudfront/v4/home.
2. Seleccione la distribución que desea deshabilitar y elija Disable (Deshabilitar).
3. Cuando se le indique que confirme, seleccione Yes, Disable (Sí, deshabilitar).

4. Seleccione la distribución desactivada y después Delete (Eliminar).
5. Cuando se le indique que confirme, seleccione Yes, Delete.

Paso 2: eliminar la zona alojada en Route 53

Antes de eliminar la zona alojada, debe eliminar los registros que creó. No es necesario que elimine los registros NS o SOA, ya que se eliminarán automáticamente cuando elimine la zona alojada.

Para eliminar el conjunto de registros

1. Abra la consola de Route 53 en https://console.aws.amazon.com/route53/.
2. En la lista de nombres de dominio, seleccione su nombre de dominio y después Go to Record Sets (Ir a conjuntos de registros).
3. En la lista de conjuntos de registros, seleccione los registros A que haya creado.

   El tipo de cada conjunto de registro está detallado en la columna Type (Tipo).
4. Seleccione la opción Delete Record Set (Eliminar conjunto de registros).
5. Cuando deba confirmar la selección, haga clic en Confirm (Confirmar).

Para eliminar una zona alojada de Route 53, realice las siguientes acciones:

1. Al finalizar el procedimiento anterior, seleccione la opción Back to Hosted Zones (Volver a zonas alojadas).
2. Seleccione el nombre de su dominio y, después, seleccione Delete Hosted Zone (Eliminar zona alojada).
3. Cuando deba confirmar la selección, haga clic en Confirm (Confirmar).

Paso 3: deshabilitar el registro y eliminar el bucket de S3

Antes de eliminar su bucket de S3, asegúrese de que la función de registro esté desactivada para el bucket. De lo contrario, AWS seguirá escribiendo registros en el bucket mientras lo elimina.

Para deshabilitar el registro en el bucket

1. Abra la consola de Amazon S3 en https://console.aws.amazon.com/s3.
2. En Buckets, elija el nombre del bucket y, a continuación, Propiedades.

3. En Properties (Propiedades), elija Logging (Registro).
4. Elimine la selección del recuadro Enabled (Habilitado).
5. Seleccione Guardar.

Ahora ya puede eliminar el bucket. Para obtener más información, consulte Eliminar un bucket.

Creación, configuración y trabajo con buckets de Amazon S3

Para almacenar datos en Amazon S3, trabaja con recursos conocidos como buckets y objetos. Un bucket es un contenedor de objetos. Un objeto es un archivo y cualquier metadato que describa ese archivo.

Para almacenar un objeto en Amazon S3, cree un bucket y, a continuación, cargue el objeto en el bucket. Cuando el objeto está en el bucket, puede abrirlo, descargarlo y moverlo. Cuando ya no necesite un objeto o un bucket, puede limpiar los recursos.

> **Note**
>
> Para obtener más información sobre el uso de la clase de almacenamiento Amazon S3 Express One Zone con buckets de directorio, consulte ¿Qué es S3 Express One Zone? y Buckets de directorio.

> **Note**
>
> Con Amazon S3 paga únicamente por lo que usa. Para obtener más información acerca de las características y precios de Amazon S3, consulte Amazon S3. Si es cliente nuevo de Amazon S3, puede comenzar con Amazon S3 de forma gratuita. Para obtener más información, consulte Capa gratuita de AWS.

Los temas de esta sección proporcionan una descripción general del trabajo con buckets en Amazon S3. Incluyen información acerca de nombrar, crear, acceder y eliminar buckets. Para obtener más información sobre la visualización o el listado de objetos de un bucket, consulte Organizar, describir y trabajar con los objetos.

Temas

- Visualización de las propiedades para un bucket de S3
- Vaciar un bucket
- Eliminar un bucket
- Establecer el comportamiento del cifrado predeterminado del lado del servidor para los buckets de Amazon S3
- Uso de Mountpoint para Amazon S3
- Configuración de transferencias de archivos rápidas y seguras con Amazon S3 Transfer Acceleration
- Utilización de buckets de pago por solicitante para transferencias de almacenamiento y uso
- Cuotas, restricciones y limitaciones de bucket

Descripción general de los buckets

Para cargar sus datos (fotos, videos, documentos, etc.) en Amazon S3, primero tiene que crear un bucket de S3 en una de las Regiones de AWS.

Un bucket es un contenedor para objetos almacenados en Amazon S3. Puede almacenar cualquier cantidad de objetos en un bucket y puede tener hasta 100 buckets en su cuenta. Para solicitar un aumento, visite la Consola de Service Quotas.

Cada objeto está almacenado en un bucket. Por ejemplo, si el objeto denominado `photos/puppy.jpg` se almacena en el bucket `amzn-s3-demo-bucket` en la región Oeste de EE. UU. (Oregón), se puede redirigir con la URL `https://amzn-s3-demo-bucket.s3.us-west-2.amazonaws.com/photos/puppy.jpg`. Para obtener más información, consulte Acceso bucket.

En términos de implementación, los buckets y los objetos son recursos de AWS, y Amazon S3 proporciona API para que pueda administrarlos. Por ejemplo, puede crear un bucket y cargar objetos con la API de Amazon S3. También puede usar la consola de Amazon S3 para realizar estas operaciones. La consola utiliza las API de Amazon S3 para enviar solicitudes a Amazon S3.

En esta sección se describe cómo trabajar con buckets. Para obtener información acerca del uso de objetos, consulte Información general de los objetos de Amazon S3.

Amazon S3 admite buckets globales, lo que significa que cada nombre de bucket debe ser único en todas las Cuentas de AWS de todas las Regiones de AWS dentro de una partición. Una partición es

una agrupación de regiones. AWS actualmente tiene tres particiones: `aws` (regiones estándar), `aws-cn` (regiones de China) y `aws-us-gov` (AWS GovCloud (US)).

Una vez que se crea un bucket, ninguna otra Cuenta de AWS de ninguna otra partición puede utilizar el nombre de ese bucket hasta que este se elimine. No debe confiar en convenciones específicas de nomenclatura de buckets para propósitos de verificación de la seguridad o disponibilidad. Para conocer las directrices de nomenclatura de buckets, consulte Reglas de nomenclatura de buckets.

Amazon S3 crea buckets en la región que usted especifique. Elija cualquier Región de AWS que esté geográficamente cerca de usted para optimizar la latencia, minimizar los costos o satisfacer los requisitos normativos. Por ejemplo, si vive en Europa, puede resultarle más conveniente crear buckets en las regiones de UE (Irlanda) o UE (Fráncfort). Para ver una lista de las regiones de Amazon S3, consulte Regiones y puntos de enlace en la Referencia general de AWS.

> **Note**
>
> Para obtener más información sobre el uso de la clase de almacenamiento Amazon S3 Express One Zone con buckets de directorio, consulte ¿Qué es S3 Express One Zone? y Buckets de directorio.

> **Note**
>
> Los objetos que pertenecen a un bucket que se crea en una Región de AWS específica nunca salen de ella, a menos que se transfieran de manera explícita a otra región. Por ejemplo, los objetos almacenados en la región UE (Irlanda) nunca salen de ella.

Temas

- Acerca de los permisos
- Administración del acceso público a los buckets
- Opciones de configuración de buckets

Acerca de los permisos

Puede utilizar las credenciales de Usuario raíz de la cuenta de AWS para crear un bucket y realizar cualquier otra operación de Amazon S3. Sin embargo, le recomendamos no utilizar las credenciales

de usuario raíz de su Cuenta de AWS para realizar solicitudes, como crear un bucket. En su lugar, cree un usuario de AWS Identity and Access Management (IAM) y concédale acceso completo a dicho usuario (de forma predeterminada, los usuarios no tienen permisos).

Estos usuarios se denominan administradores. Para interactuar con AWS y realizar tareas, tales como crear un bucket, crear usuarios y concederles permisos, puede utilizar las credenciales de usuario administrador en lugar de las credenciales de usuario raíz de su cuenta.

Para obtener más información, consulte Credenciales de Usuario raíz de la cuenta de AWS y credenciales de usuario de IAM en la Referencia general de AWS y las Prácticas recomendadas de seguridad en IAM en la Guía del usuario de IAM.

La Cuenta de AWS que crea un recurso se convierte en la propietaria de este recurso. Por ejemplo, si usted crea un usuario de IAM en su Cuenta de AWS y le concede al usuario permiso para crear un bucket, el usuario puede crear un bucket. Pero el usuario no es el propietario del bucket, sino la Cuenta de AWS a la que pertenece el usuario. El usuario necesita un permiso adicional del propietario del recurso para realizar otras operaciones relacionadas con el bucket. Para obtener más información acerca de la administración de permisos para los recursos de Amazon S3, consulte Administración de identidades y accesos para Amazon S3.

Administración del acceso público a los buckets

El acceso público se otorga a buckets y objetos a través de listas de control de acceso (ACL), políticas de bucket o ambas. Para ayudarlo a administrar el acceso público a los recursos de Amazon S3, Amazon S3 proporciona configuraciones para bloquear el acceso público. La configuración de bloqueo de acceso público de Amazon S3 puede anular políticas de buckets y ACL para que pueda aplicar al acceso público límites uniformes a dichos recursos. Puede aplicar una configuración de bloqueo de acceso público a buckets individuales o a todos los buckets de la cuenta.

Para ayudar a garantizar que todos los buckets y objetos de Amazon S3 tengan el acceso público bloqueado, los cuatro ajustes del bloqueo de acceso público se activan de forma predeterminada al crear un nuevo bucket. Le recomendamos que también active los cuatro ajustes del bloqueo de acceso público de la cuenta. Estos ajustes bloquean la totalidad del acceso público a todos los buckets actuales y futuros.

Antes de aplicar estos ajustes, verifique que sus aplicaciones funcionen correctamente sin acceso público. Si necesita algún nivel de acceso público a los buckets u objetos, como, por ejemplo, para alojar un sitio web estático, tal como se describió en Alojamiento de un sitio web estático mediante Amazon S3, puede personalizar los ajustes individuales para que se adapten a sus casos de uso

de almacenamiento. Para obtener más información, consulte Bloquear el acceso público a su almacenamiento de Amazon S3.

Sin embargo, recomendamos encarecidamente mantener activado el bloqueo de acceso público. Si desea conservar las cuatro configuraciones de bloqueo de acceso público activados y alojar un sitio web estático, puede utilizar el control de acceso de origen (OAC) de Amazon CloudFront. Amazon CloudFront proporciona las capacidades necesarias para configurar un sitio web estático seguro. Los sitios web estáticos de Amazon S3 solo admiten puntos de conexión HTTP. Amazon CloudFront utiliza el almacenamiento duradero de Amazon S3 a la vez que proporciona encabezados de seguridad adicionales, como HTTPS. HTTPS agrega seguridad al cifrar una solicitud HTTP normal y proteger contra ataques cibernéticos comunes.

Para obtener información, consulte Introducción a un sitio web estático seguro en la guía para desarrolladores de Amazon CloudFront.

> **Note**
>
> Si ve un `Error` cuando muestre los buckets y la configuración de acceso público, es posible que no tenga los permisos necesarios. Asegúrese de que los siguientes permisos se hayan agregado a la política del usuario o rol:
>
> ```
> s3:GetAccountPublicAccessBlock
> s3:GetBucketPublicAccessBlock
> s3:GetBucketPolicyStatus
> s3:GetBucketLocation
> s3:GetBucketAcl
> s3:ListAccessPoints
> s3:ListAllMyBuckets
> ```
>
> En algunos casos excepcionales, las solicitudes también pueden producir un error debido a una interrupción de Región de AWS.

Opciones de configuración de buckets

Amazon S3 admite varias opciones para que configure su bucket. Por ejemplo, puede configurar su bucket para alojamiento de sitios web, agregar configuraciones para administrar el ciclo de vida de los objetos en el bucket y configurar el bucket para registrar la totalidad del acceso al bucket. Amazon S3 admite subrecursos para que almacene y administre la información de configuración del

bucket. Puede usar la API de Amazon S3 para crear y administrar estos subrecursos. Sin embargo, también puede usar la consola o los SDK de AWS.

> **Note**
>
> También hay configuraciones a nivel del objeto. Por ejemplo, puede configurar permisos a nivel del objeto mediante la configuración de una Access Control List (ACL, Lista de control de acceso) que sea específica para ese objeto.

Estos se denominan subrecursos porque existen en el contexto de un bucket u objeto específico. En la siguiente tabla, se muestran los subrecursos que le permiten administrar configuraciones específicas de buckets.

Subrecurso	Descripción
cors (uso compartido de recursos entre orígenes)	Puede configurar su bucket para permitir solicitudes entre orígenes. Para obtener más información, consulte Uso compartido de recursos entre orígenes (CORS).
notificación de eventos	Puede habilitar su bucket para que le envíe notificaciones de eventos especificados de buckets. Para obtener más información, consulte Notificaciones de eventos de Amazon S3.
ciclo de vida	Puede definir reglas de ciclo de vida para los objetos de su bucket que tienen un ciclo de vida bien definido. Por ejemplo, puede definir una regla para archivar objetos un año después de su creación o para eliminar objetos 10 años después de su creación. Para obtener más información, consulte Administración del ciclo de vida del almacenamiento.
location	Cuando crea un bucket, especifica la Región de AWS en la que desea que Amazon S3 cree el bucket. Amazon S3 almacena esta información

Subrecurso	Descripción
	en el subrecurso ubicación y proporciona una API para que recupere esta información.
registro	El registro le permite realizar un seguimiento de las solicitudes de acceso a su bucket. Cada entrada del registro de acceso contiene detalles de la solicitud de acceso tales como el solicitante, el nombre del bucket, la hora de la solicitud, la acción solicitada, el estado de la respuesta y el código de error, si hay alguno. La información del registro de acceso puede ser útil en auditorías de acceso y seguridad. También puede ayudarle a conocer mejor su base de clientes y entender su factura de Amazon S3. Para obtener más información, consulte Registro de solicitudes con registro de acceso al servidor.
bloqueo de objetos	Para usar Bloqueo de objetos de S3, debe habilitarlo para un bucket. De manera opcional, puede configurar un modo y un periodo de retención predeterminados para los nuevos objetos colocados en el bucket. Para obtener más información, consulte Usar Bloqueo de objetos de S3.
política y ACL (lista de control de acceso)	Todos sus recursos (como los buckets y objetos) son de carácter privado de forma predeterminada. Amazon S3 admite opciones de política y lista de control de acceso (ACL) de buckets para que conceda y administre los permisos a nivel del bucket. Amazon S3 almacena la información de los permisos en los subrecursos política y ACL. Para obtener más información, consulte Administración de identidades y accesos para Amazon S3.
replicación	La reproducción consiste en la copia automática y asíncrona de los objetos de los buckets ubicados en la misma o en diferentes Regiones de AWS. Para obtener más información, consulte Información general de la replicación de objetos.

Subrecurso	Descripción
requestPayment	De forma predeterminada, la Cuenta de AWS que crea el bucket (la propietaria del bucket) paga las descargas realizadas desde el bucket. Mediante este subrecurso, el propietario del bucket puede especificar que se le cobre la descarga a la persona que la solicita. Amazon S3 proporciona una API para que administre este subrecurso. Para obtener más información, consulte Utilización de buckets de pago por solicitante para transferencias de almacenamiento y uso.
etiquetado	Puede agregar etiquetas de asignación de costos a su bucket para clasificar en categorías los costos de AWS y realizar un seguimiento de ellos. Amazon S3 proporciona el subrecurso etiquetado para almacenar y administrar las etiquetas en un bucket. Mediante las etiquetas que se aplican a su bucket, AWS genera un informe de asignación de costos con el uso y los costos agregados por sus etiquetas. Para obtener más información, consulte Informes de facturación y uso de Amazon S3.
aceleración de transferencia	Transfer Acceleration permite transferir archivos de forma rápida, fácil y segura entre su cliente y un bucket de S3 a larga distancia. Transfer Acceleration aprovecha las ubicaciones de borde de Amazon CloudFront distribuidas globalmente. Para obtener más información, consulte Configuración de transferencias de archivos rápidas y seguras con Amazon S3 Transfer Acceleration.
control de versiones	EL control de versiones lo ayuda a recuperar objetos que se han sobrescrito y eliminado accidentalmente. Recomendamos el control de versiones como práctica recomendada para recuperar objetos que se han eliminado o sobrescrito accidentalmente. Para obtener más información, consulte Usar el control de versiones en buckets de S3.

Subrecurso	Descripción
sitio web	Puede configurar su bucket para el alojamiento de sitios web estáticos. Amazon S3 almacena esta configuración mediante la creación del subrecurs o sitio web. Para obtener más información, consulte Alojamiento de un sitio web estático mediante Amazon S3.

Reglas de nomenclatura de buckets

Las siguientes reglas se aplican a la nomenclatura de buckets de uso general y buckets de directorio en Amazon S3:

Temas

- Reglas de nomenclatura de los buckets de uso general
- Reglas de nomenclatura de buckets de directorio

Reglas de nomenclatura de los buckets de uso general

Las siguientes reglas de nomenclatura se aplican para buckets de uso general.

- Los nombres de bucket deben tener entre 3 caracteres (mín.) y 63 caracteres (máx.).
- Los nombres de bucket pueden consistir únicamente de letras minúsculas, números, puntos (.) y guiones (-).
- Los nombres de bucket deben comenzar y terminar con una letra o un número.
- Los nombres de bucket no deben contener dos puntos adyacentes.
- Los nombres de buckets no deben tener el formato de una dirección IP (por ejemplo, 192.168.5.4).
- Los nombres de los buckets no deben comenzar con el prefijo `xn--`.
- Los nombres de los buckets no deben comenzar con el prefijo `sthree-`.
- Los nombres de los buckets no deben comenzar con el prefijo `sthree-configurator`.
- Los nombres de los buckets no deben comenzar con el prefijo `amzn-s3-demo-`.

- Los nombres de los buckets no deben terminar con el sufijo `-s3alias`. Este sufijo está reservado para nombres de alias de punto de acceso. Para obtener más información, consulte Uso de un alias de estilo de bucket para su punto de acceso de bucket de S3.
- Los nombres de los buckets no deben terminar con el sufijo `--ol-s3`. Este sufijo está reservado para nombres de alias de punto de acceso de Object Lambda. Para obtener más información, consulte Cómo usar un alias de estilo de bucket para su punto de acceso de Object Lambda de bucket de S3.
- Los nombres de los buckets no deben terminar con el sufijo `.mrap`. Este sufijo está reservado para nombres de punto de acceso de varias regiones. Para obtener más información, consulte Reglas para asignar nombres a los puntos de acceso de varias regiones de Amazon S3.
- Los nombres de los buckets no deben terminar con el sufijo `--x-s3`. Este sufijo está reservado para buckets de directorio. Para obtener más información, consulte Reglas de nomenclatura de buckets de directorio.
- Los nombres de los buckets deben ser únicos en todas las Cuentas de AWS de todas las Regiones de AWS de una partición. Una partición es una agrupación de regiones. AWS actualmente tiene tres particiones: `aws` (regiones estándar), `aws-cn` (regiones de China) y `aws-us-gov` (AWS GovCloud (US)).
- Otra Cuenta de AWS de la misma partición no puede utilizar el mismo nombre de bucket hasta que se elimine el bucket.
- Los buckets utilizados con Amazon S3 Transfer Acceleration no pueden tener puntos (.) en sus nombres. Para obtener más información acerca de Transfer Acceleration, consulte Configuración de transferencias de archivos rápidas y seguras con Amazon S3 Transfer Acceleration.

Para obtener una mejor compatibilidad, se recomienda evitar el uso de puntos (.) en los nombres de los buckets, excepto para los buckets que se utilizan únicamente para el alojamiento estático de sitios web. Si incluye puntos en el nombre de un bucket, no puede usar direccionamiento de estilo host virtual a través de HTTPS, a menos que realice su propia validación de certificado. Esto se debe a que los certificados de seguridad utilizados para el alojamiento virtual de los buckets no funcionan para los buckets con puntos en sus nombres.

Esta limitación no afecta a los buckets utilizados para el alojamiento de sitios web estáticos, ya que el alojamiento de sitios web estáticos solo está disponible a través de HTTP. Para obtener más información acerca del direccionamiento de tipo de host virtual, consulte Alojamiento virtual de buckets. Para obtener más información sobre el alojamiento estático de sitios web, consulte Alojamiento de un sitio web estático mediante Amazon S3.

> **Note**
>
> Antes del 1 de marzo de 2018, los buckets creados en la región EE. UU. Este (Norte de Virginia) podían tener nombres de hasta 255 caracteres e incluir letras mayúsculas y guiones bajos. A partir del 1 de marzo de 2018, los nuevos buckets de EE. UU. Este (Norte de Virginia) deben ajustarse a las mismas reglas aplicadas en todas las demás regiones.

Para obtener información acerca de los nombres de clave de objeto, consulte Creación de nombres de clave de objeto.

Nombres de buckets de uso general de ejemplo

Los nombres de bucket de ejemplo siguientes son válidos y siguen las pautas de nomenclatura recomendadas para buckets de uso general:

- `docexamplebucket1`
- `log-delivery-march-2020`
- `my-hosted-content`

Los nombres de bucket de ejemplo siguientes son válidos pero no se recomiendan para usos distintos del alojamiento estático de sitios web:

- `docexamplewebsite.com`
- `www.docexamplewebsite.com`
- `my.example.s3.bucket`

Los nombres de bucket de ejemplo siguientes no son válidos:

- `doc_example_bucket` (contiene guiones bajos)
- `DocExampleBucket` (contiene letras mayúsculas)
- `doc-example-bucket-` (termina con un guion)

Reglas de nomenclatura de buckets de directorio

Las siguientes reglas de nomenclatura se aplican a los buckets de directorio.

- Ser únicos dentro de la Región de AWS y la zona de disponibilidad elegida.
- El nombre debe tener entre 3 (mín.) y 63 caracteres (máx.), incluido el sufijo.
- Constar de letras minúsculas, números y guiones (-).
- Comenzar y terminar por un número o una letra.
- Debe incluir el siguiente sufijo: `--`*`azid`*`--x-s3`.
- Los nombres de los buckets no deben comenzar con el prefijo `xn--`.
- Los nombres de los buckets no deben comenzar con el prefijo `sthree-`.
- Los nombres de los buckets no deben comenzar con el prefijo `sthree-configurator`.
- Los nombres de los buckets no deben comenzar con el prefijo `amzn-s3-demo-`.
- Los nombres de los buckets no deben terminar con el sufijo `-s3alias`. Este sufijo está reservado para nombres de alias de punto de acceso. Para obtener más información, consulte Uso de un alias de estilo de bucket para su punto de acceso de bucket de S3.
- Los nombres de los buckets no deben terminar con el sufijo `--ol-s3`. Este sufijo está reservado para nombres de alias de punto de acceso de Object Lambda. Para obtener más información, consulte Cómo usar un alias de estilo de bucket para su punto de acceso de Object Lambda de bucket de S3.
- Los nombres de los buckets no deben terminar con el sufijo `.mrap`. Este sufijo está reservado para nombres de punto de acceso de varias regiones. Para obtener más información, consulte Reglas para asignar nombres a los puntos de acceso de varias regiones de Amazon S3.

> **Note**
>
> Al crear un bucket de directorio mediante la consola, se agrega automáticamente un sufijo al nombre base que proporcione. Este sufijo incluye el ID de zona de disponibilidad de la que haya elegido.
> Al crear un bucket de directorio mediante una API, en la solicitud debe proporcionar el sufijo completo, incluido el ID de la zona de disponibilidad. Para obtener una lista de los ID de las zonas de disponibilidad, consulte Zonas y regiones de disponibilidad de S3 Express One Zone.

Acceso y publicación de un bucket de Amazon S3

Para publicar sus buckets de Amazon S3 y acceder a ellos, puede utilizar varias herramientas. Revise las siguientes herramientas para determinar qué enfoque se adapta a su caso de uso:

- Consola de Amazon S3: con la consola de Amazon S3, puede acceder con facilidad a un bucket y modificar sus propiedades. Con la IU de la consola, puede realizar la mayoría de las operaciones en el bucket sin tener que escribir ningún código.
- AWS CLI: si necesita acceder a varios buckets, puede ahorrar tiempo utilizando la CLI de AWS Command Line Interface (AWS CLI) para automatizar tareas comunes y repetitivas. La automatización mediante scripts y la repetibilidad de las acciones comunes son consideraciones frecuentes a medida que las organizaciones crecen. Para obtener más información, consulte Desarrollo con Amazon S3 mediante la AWS CLI.
- API de REST de Amazon S3: puede usar la API de REST de Amazon S3 para escribir sus propios programas y acceder al bucket mediante programación. Amazon S3 admite una arquitectura de la API en la que los buckets y objetos son recursos, cada uno con un URI (Identificador uniforme de recursos) mediante el cual se identifica específicamente el recurso. Para obtener más información, consulte Desarrollo con Amazon S3 mediante la API REST.

Según el caso de uso de su bucket de Amazon S3, se recomiendan diferentes métodos para acceder a los datos subyacentes de sus buckets. La siguiente lista incluye casos de uso comunes para acceder a sus datos.

- Sitios web estáticos: puede usar Amazon S3 para alojar un sitio web estático. En este caso de uso, puede configurar su bucket de S3 para que funcione como un sitio web. Para ver un ejemplo en el que se muestran los pasos para el alojamiento de un sitio web en Amazon S3, consulte Tutorial: configuración de un sitio web estático en Amazon S3.

  Para alojar un sitio web estático con una configuración de seguridad como Bloquear acceso público habilitada, le recomendamos utilizar Amazon CloudFront con Control de acceso de origen (OAC) e implementar encabezados de seguridad adicionales, como HTTPS. Para obtener más información, consulte Introducción a un sitio web seguro estático.

  > **Note**
  >
  > Amazon S3 admite URL tanto de tipo alojamiento virtual como de tipo ruta para obtener acceso a sitios web estáticos. Debido a que se puede obtener acceso a los buckets mediante los URL de tipo ruta y alojamiento virtual, le recomendamos crear buckets con

nombres de buckets compatibles con DNS. Para obtener más información, consulte Cuotas, restricciones y limitaciones de bucket.

- Conjuntos de datos compartidos: a medida que escala en Amazon S3, es habitual adoptar un modelo de varios inquilinos, en el que se asignan distintos clientes finales o unidades de negocio a prefijos únicos dentro de un bucket compartido. Al utilizar los puntos de acceso de Amazon S3, puede dividir una política de bucket grande en políticas de puntos de acceso independientes y discretas para cada aplicación que necesite acceder al conjunto de datos compartido. Este enfoque facilita centrarse en crear la política de acceso adecuada para una aplicación sin interrumpir lo que hace cualquier otra aplicación dentro del conjunto de datos compartido. Para obtener más información, consulte Administración del acceso a datos con puntos de acceso de Amazon S3.
- Cargas de trabajo de alto rendimiento: Mountpoint para Amazon S3 es un cliente de archivos de código abierto de alto rendimiento para montar un bucket de Amazon S3 como un sistema de archivos local. Con Mountpoint, sus aplicaciones pueden acceder a objetos almacenados en Amazon S3 mediante operaciones del sistema de archivos, como abrir y leer. Mountpoint traduce automáticamente estas operaciones en llamadas a la API de objetos de S3, lo que proporciona a sus aplicaciones acceso al almacenamiento elástico y al rendimiento de Amazon S3 a través de una interfaz de archivos. Para obtener más información, consulte Uso de Mountpoint para Amazon S3.
- Aplicaciones multirregionales: los puntos de acceso multirregionales de Amazon S3 proporcionan un punto de conexión global que las aplicaciones pueden usar para atender solicitudes de buckets de S3 que se encuentran en varias Regiones de AWS. Puede utilizar puntos de acceso multirregional para crear aplicaciones multirregionales con la misma arquitectura que se utiliza en una sola región y, a continuación, ejecutar esas aplicaciones en cualquier parte del mundo. En lugar de enviar solicitudes a través de Internet público, los puntos de acceso de varias regiones proporcionan resistencia de red integrada con la aceleración de las solicitudes basadas en Internet a Amazon S3. Para obtener más información, consulte Puntos de acceso de varias regiones de Amazon S3.
- Creación de nuevas aplicaciones: puede utilizar los SDK de AWS para desarrollar aplicaciones con Amazon S3. Los SDK de AWS simplifican las tareas de programación dado que incluyen la API de REST de Amazon S3 subyacente. Para crear aplicaciones web y móviles, puede usar los Mobile SDK de AWS y la biblioteca de el bucket de JavaScript AWS Amplify. Para obtener más información, consulte Desarrollo con Amazon S3 mediante los SDK de AWS.
- Protocolo de transferencia de archivos (SFTP) Secure Shell (SSH): si intenta transferir datos confidenciales de forma segura a través de Internet, puede utilizar un servidor habilitado

para SFTP con su bucket de Amazon S3. AWS SFTP es un protocolo de red que admite la funcionalidad completa de seguridad y autenticación de SSH. Con este protocolo, tiene un control detallado sobre la identidad, los permisos y las claves de los usuarios, o puede utilizar las políticas de IAM para gestionar el acceso. Para asociar un servidor habilitado para SFTP a su bucket de Amazon S3, asegúrese de crear primero su servidor habilitado para SFTP. A continuación, debe configurar cuentas de usuario y asociar el servidor con un bucket de Amazon S3. Para una explicación de este proceso, consulte AWS Transfer for SFTP – Servicio totalmente administrado para Amazon S3 en AWS Blogs.

Obtención de una lista de buckets

Para enumerar todos sus buckets, debe tener el permiso de `s3:ListAllMyBuckets`. Para acceder a un bucket, asegúrese de obtener también los permisos de AWS Identity and Access Management (IAM) necesarios para mostrar el contenido del bucket especificado. Para una política de bucket de ejemplo que conceda acceso a un bucket de S3, consulte Permiso para que el usuario de IAM tenga acceso a uno de los buckets. Si encuentra un error de acceso HTTP denegado (403 Prohibido), consulte Políticas de bucket y de IAM.

Puede mostrar su bucket mediante la consola de Amazon S3, la AWS o los SDK de AWS CLI.

Uso de la consola de S3

1. Inicie sesión en la AWS Management Console y abra la consola de Amazon S3 en https://console.aws.amazon.com/s3/.
2. En el panel de navegación izquierdo, elija Instancias.
3. En la lista General purpose buckets (Buckets de uso general), elija el nombre del bucket que desea ver.

> **Note**
>
> La lista Buckets de uso general incluye los buckets que se encuentran en todas las Regiones de AWS.

Uso de la AWS CLI

Para usar la AWS CLI para acceder a un bucket de S3 o generar un listado de buckets de S3, utilice el comando `ls`. Cuando publique todos los objetos de su bucket, tenga en cuenta que debe tener el permiso `s3:ListBucket`.

Para usar el comando de ejemplo, sustituya *DOC-EXAMPLE-BUCKET1* por el nombre de su bucket.

```
$ aws s3 ls s3://DOC-EXAMPLE-BUCKET1
```

El siguiente comando de ejemplo muestra todos los buckets de Amazon S3 en su cuenta:

```
$ aws s3 ls
```

Para obtener más información y ejemplos, consulte Enumerar buckets y objetos.

Uso de los AWS SDK

También puede acceder a un bucket de Amazon S3 mediante la operación de la API de ListBuckets. Para ver ejemplos de cómo utilizar esta operación con diferentes SDK de AWS, consulte Uso de `ListBuckets` con un AWS SDK o la CLI.

Crear un bucket

Para cargar los datos en Amazon S3, primero debe crear un bucket de Amazon S3 en una de las Regiones de AWS. Al crear un bucket, debe elegir un nombre de bucket y una región. Si lo desea, puede elegir otras opciones de administración de almacenamiento para el bucket. Después de crear un bucket, no se puede cambiar su nombre ni su región. Para obtener información acerca de la nomenclatura de los buckets, consulte Reglas de nomenclatura de buckets.

La Cuenta de AWS que crea el bucket es propietaria de este. Puede cargar la cantidad de objetos que desee en el bucket. De forma predeterminada, puede crear hasta 100 buckets en cada una de sus Cuentas de AWS. Si necesita más buckets, puede presentar una solicitud de aumento del límite del servicio para aumentar el límite de buckets de la cuenta hasta un máximo de 1000 buckets. Para obtener información acerca de cómo enviar un aumento del límite de buckets, consulte el tema sobre las cuotas de Servicio de AWS en la Referencia general de AWS. En un bucket, puede almacenar la cantidad de objetos que desee.

S3 Object Ownership es una configuración de bucket de Amazon S3 que puede usar para controlar la propiedad de los objetos que se cargan en el bucket y para activar o desactivar las listas de control

de acceso (ACL). De forma predeterminada, la propiedad de objetos se establece en la configuración impuesta por el propietario del bucket. Además, todas las ACL están deshabilitadas. Cuando las ACL están desactivadas, el propietario del bucket posee todos los objetos del bucket y administra el acceso a los datos de forma exclusiva mediante políticas.

Para obtener más información, consulte Control de la propiedad de los objetos y desactivación de las ACL del bucket.

El cifrado del servidor con claves administradas de Amazon S3 (SSE-S3) es el nivel básico de la configuración de cifrado para cada bucket de Amazon S3. Todos los objetos nuevos cargados en un bucket de S3 se cifran de forma automática con SSE-S3 como el nivel básico de la configuración de cifrado. Si desea utilizar un tipo de cifrado predeterminado distintos, también puede especificar el cifrado del servidor con claves de AWS Key Management Service (AWS KMS) (SSE-KMS) o claves proporcionadas por el cliente (SSE-C) para cifrar los datos. Para obtener más información, consulte Establecer el comportamiento del cifrado predeterminado del lado del servidor para los buckets de Amazon S3.

Para crear un bucket, puede utilizar la consola de Amazon S3, las API de Amazon S3, la AWS CLI o los SDK de AWS. Para obtener más información acerca de los permisos necesarios para crear un bucket, consulte CreateBucket en la Referencia de la API de Amazon Simple Storage Service.

Uso de la consola de S3

1. Inicie sesión en la AWS Management Console y abra la consola de Amazon S3 en https://console.aws.amazon.com/s3/.
2. En la barra de navegación de la parte superior de la página, elija el nombre de la Región de AWS que aparece. A continuación, elija en Region (Región) la región en la que desea crear un bucket.

   > **Note**
   >
   > Puede seleccionar una región cercana para minimizar la latencia y los costos, así como para satisfacer los requisitos normativos. Los objetos almacenados en una región nunca abandonarán esa región salvo que usted los transfiera de forma específica a otra. Para una lista de Regiones de AWS de Amazon S3, consulte Puntos de conexión de Servicio de AWS en la Referencia general de Amazon Web Services.

3. En el panel de navegación izquierdo, elija Instancias.

4. Elija Crear bucket.

   Se abrirá la página Crear bucket.

5. En Configuración general, vea la Región de AWS donde se creará el bucket.
6. En Tipo de depósito, seleccione Uso general.
7. En Nombre del bucket, escriba un nombre para el bucket.

   El nombre del bucket debe:

   - Ser exclusivo dentro de una partición. Una partición es una agrupación de regiones. AWS actualmente tiene tres particiones: `aws` (regiones estándar), `aws-cn` (regiones de China) y `aws-us-gov` (AWS GovCloud (US) Regions).
   - Tener entre 3 y 63 caracteres.
   - Consistir únicamente de letras minúsculas, números, puntos (.) y guiones (-). Para obtener una mejor compatibilidad, se recomienda evitar el uso de puntos (.) en los nombres de los buckets, excepto para los buckets que se utilizan únicamente para el alojamiento estático de sitios web.
   - Comenzar y terminar por un número o una letra.

   Una vez que haya creado el bucket, no podrá modificar su nombre. Para obtener más información sobre la nomenclatura de los buckets, consulte Reglas de nomenclatura de buckets.

   > **Important**
   >
   > Evite incluir información confidencial, como números de cuenta, en el nombre del bucket. El nombre del bucket será visible en las URL que señalan a los objetos almacenados en él.

8. AWS Management Console le permite copiar la configuración de un bucket existente en el nuevo bucket. Si no desea copiar la configuración de un bucket existente, vaya al paso siguiente.

   > **Note**
   >
   > Esta opción:
   >
   > - No está disponible en la AWS CLI y solo está disponible en la consola
   > - No está disponible para buckets de directorio

- No copia la política de bucket del bucket existente al nuevo bucket

Para copiar la configuración de un bucket existente, en Copiar la configuración del depósito existente, seleccione Elegir bucket. Se abre la ventana Elegir bucket. Busque el bucket con los ajustes que quiera copiar y seleccione Elegir bucket. Se cierra la ventana Elegir bucket y se vuelve a abrir la ventana Crear bucket.

En Copiar la configuración del bucket existente, ahora verá el nombre del bucket que ha seleccionado. También verá la opción Restaurar los valores predeterminados que puede usar para eliminar la configuración del bucket copiada. Revise la configuración restante del bucket en la página Crear bucket. Verá que ahora coinciden con la configuración del bucket que seleccionó. Puede saltar al paso final.

9. En Propiedad de objetos, para desactivar o habilitar las ACL y controlar la propiedad de los objetos cargados en el bucket, elija una de las siguientes configuraciones:

   ACL desactivadas

   - Propietario del bucket obligatorio (predeterminado): las ACL están desactivadas y el propietario del bucket tiene automáticamente la propiedad y el control total sobre cada objeto del bucket. Las ACL ya no afectan a los permisos de acceso de los datos del bucket de S3. El bucket utiliza políticas exclusivamente para definir el control de acceso.

     Las ACL están desactivadas de forma predeterminada. La mayoría de los casos de uso modernos de Amazon S3 ya no requieren el uso de ACL. Le recomendamos que mantenga las ACL desactivadas, excepto en circunstancias inusuales en las que necesite controlar el acceso a cada objeto de manera individual. Para obtener más información, consulte Control de la propiedad de los objetos y desactivación de las ACL del bucket.

   ACL habilitadas

   - Propietario del bucket preferido: el propietario del bucket tiene la propiedad y el control total sobre los nuevos objetos que otras cuentas escriben en el bucket con la ACL predefinida `bucket-owner-full-control`.

     Si aplica la configuración de propietario del bucket preferido para requerir que todas las cargas de Amazon S3 incluyan la ACL predefinida `bucket-owner-full-control`, puede agregar una política de bucket que solo permita cargas de objetos que utilicen esta ACL.

- Escritor del objeto: la Cuenta de AWS que carga un objeto es propietaria del objeto, tiene control total sobre él y puede conceder a otros usuarios acceso a él a través de ACL.

> **Note**
>
> La configuración predeterminada es Aplicada al propietario del bucket. Para aplicar la configuración predeterminada y mantener las ACL deshabilitadas, solo se necesita el permiso `s3:CreateBucket`. Para habilitar las ACL, debe tener el permiso `s3:PutBucketOwnershipControls`.

10. En Configuración de bloqueo de acceso público para este bucket, elija la configuración Bloquear acceso público que desee aplicar al bucket.

    De forma predeterminada, las cuatro configuraciones de Bloqueo de acceso público estarán activas. Le recomendamos que deje todas las configuraciones activadas a menos que sepa que necesita desactivar una o varias para su caso de uso específico. Para obtener más información acerca del bloqueo del acceso público, consulte Bloquear el acceso público a su almacenamiento de Amazon S3.

    > **Note**
    >
    > Para habilitar todas las configuraciones de Bloqueo de acceso público, solo se requiere el permiso `s3:CreateBucket`. Para desactivar cualquier configuración de Bloqueo de acceso público, debe tener el permiso `s3:PutBucketPublicAccessBlock`.

11. (Opcional) En Control de versiones de buckets, puede elegir si desea mantener variantes de objetos en su bucket. Para obtener más información sobre el control de versiones, consulte Usar el control de versiones en buckets de S3.

    Para deshabilitar o habilitar el control de versiones en su bucket, elija Disable (Deshabilitar) o Enable (Habilitar).

12. (Opcional) En Tags (Etiquetas), puede elegir añadir etiquetas a su bucket. Las etiquetas son pares clave-valor que se utilizan para categorizar el almacenamiento de información.

    Para agregar una etiqueta de bucket, introduzca un valor en Clave y opcionalmente otro en Valor y elija Añadir etiqueta.

13. En Cifrado predeterminado, elija Editar.

14. Para configurar el cifrado predeterminado, en Tipo de cifrado, elija una de las siguientes opciones:

    - Clave administrada de Amazon S3 (SSE-S3)
    - Clave de AWS Key Management Service (SSE-KMS)

    > **Important**
    >
    > Si utiliza la opción de SSE-KMS para la configuración de cifrado predeterminado, se le aplicará la cuota de solicitudes por segundo (RPS) de AWS KMS. Para obtener más información acerca de las cuotas de AWS KMS y cómo solicitar un aumento de cuota, consulte Cuotas en la Guía para desarrolladores de AWS Key Management Service.

    Los buckets y los objetos nuevos se cifran mediante el cifrado del lado del servidor con una clave administrada de Amazon S3 como nivel básico de configuración de cifrado. Para obtener más información acerca del cifrado predeterminado, consulte Establecer el comportamiento del cifrado predeterminado del lado del servidor para los buckets de Amazon S3.

    Para obtener más información sobre el uso del cifrado del lado del servidor de Amazon S3 para cifrar los datos, consulte Uso del cifrado del servidor con claves administradas por Amazon S3 (SSE-S3).

15. Si ha elegido la clave de AWS Key Management Service (SSE-KMS), haga lo siguiente:

    a. En Clave de AWS KMS, especifique su clave de KMS de una de las siguientes maneras:

        - Para seleccionar de una lista de claves de KMS disponibles, marque Elija entre sus claves de AWS KMS keys y seleccione su clave de KMS de la lista de claves disponibles.

          En esta lista aparecen tanto la Clave administrada de AWS (`aws/s3`) como las claves administradas por el cliente. Para obtener más información acerca de las claves administradas por el cliente, consulte Claves de cliente y claves de AWS en la Guía para desarrolladores de AWS Key Management Service.
        - Para introducir el ARN de la clave de KMS, elija Introducir el ARN de la AWS KMS key e introduzca el ARN de la clave de KMS en el campo que aparece.
        - Para crear una nueva clave administrada por el cliente en la consola de AWS KMS, elija Crear una clave de KMS.

Para obtener más información acerca de cómo crear una AWS KMS key, consulte Creación de claves en la AWS Key Management Service Guía para desarrolladores.

> **Important**
>
> Solo puede utilizar las claves de KMS que estén disponibles en la misma Región de AWS del bucket. La consola de Amazon S3 solo muestra las primeras 100 claves de KMS de la misma región del bucket. Para utilizar una clave de KMS que no aparezca en la lista, debe introducir el ARN de la clave de KMS. Si desea utilizar una clave de KMS propiedad de una cuenta de diferente, primero debe tener permiso para utilizar la clave y, después, debe introducir el ARN de la clave de KMS. Para obtener más información sobre los permisos entre cuentas para las claves de KMS, consulte Crear claves de KMS que otras cuentas puedan utilizar en la Guía para desarrolladores de AWS Key Management Service. Para obtener más información sobre SSE-KMS, consulte Especificación del cifrado del lado del servidor con AWS KMS (SSE-KMS).
>
> Cuando utilice una AWS KMS key para el cifrado en el lado del servidor en Amazon S3, debe elegir una clave de cifrado de KMS simétrica. Amazon S3 admite solo claves de KMS de cifrado simétricas y no claves de KMS asimétricas. Para obtener más información, consulte Identificación de claves de KMS simétricas y asimétricas en la Guía para desarrolladores de AWS Key Management Service.

Para obtener más información acerca de cómo crear una AWS KMS key, consulte Creación de claves en la AWS Key Management Service Guía para desarrolladores. Para obtener más información acerca del uso de AWS KMS con Amazon S3, consulte Uso del cifrado del servidor con claves de AWS KMS (SSE-KMS).

b. Cuando configure el bucket para que use el cifrado predeterminado con SSE-KMS, también puede habilitar las claves de bucket de S3. Las claves de bucket de S3 reducen el costo del cifrado al reducir el tráfico de solicitudes de Amazon S3 a AWS KMS. Para obtener más información, consulte Reducción del costo de SSE-KMS con las claves de bucket de Amazon S3.

   Para utilizar las claves de bucket de S3, en Clave de bucket, seleccione Habilitar.

16. (Opcional) Si desea habilitar el bloqueo de objetos en S3, haga lo siguiente:

a. Seleccione Advanced settings (Ajustes avanzados).

> **Important**
>
> Al habilitar Bloqueo de objetos, también se habilita el control de versiones para el bucket. Después de habilitar, debe configurar la retención predeterminada de Object Lock y la configuración de retención legal para evitar que los nuevos objetos se eliminen o se sobrescriban.

b. Si desea habilitar el bloqueo de objetos, elija Enable (Habilitar), lea la advertencia que aparece y acéptela.

Para obtener más información, consulte Usar Bloqueo de objetos de S3.

> **Note**
>
> Para crear un bucket con Bloqueo de objetos, debe tener los siguientes permisos: `s3:CreateBucket`, `s3:PutBucketVersioning` y `s3:PutBucketObjectLockConfiguration`.

17. Elija Crear bucket.

Uso de la SDKs AWS

Si usa los SDK de AWS con el objeto de crear un bucket, debe crear un cliente y, luego, utilizar el cliente a fin de enviar una solicitud para crear un bucket. Como práctica recomendada, debe crear el cliente y el bucket en la misma Región de AWS. Si no especifica una región al crear un cliente o un bucket, Amazon S3 utiliza la región predeterminada, Este de EE. UU. (Norte de Virginia). Si desea restringir la creación del bucket a una Región de AWS específica, utilice la clave de condición `LocationConstraint`.

Para crear un cliente con el objeto de obtener acceso a un punto de conexión de doble pila, debe especificar una Región de AWS. Para obtener más información, consulte Puntos de enlace de doble pila. Para ver una lista de las Regiones de AWS disponibles, consulte Regiones y puntos de conexión en la Referencia general de AWS.

Al crear un cliente, la región se asigna al punto de conexión específico de la región. El cliente utiliza este punto de conexión para comunicarse con Amazon S3: `s3.region.amazonaws.com`. Si su región se lanzó después del 20 de marzo de 2019, su cliente y el bucket deben estar en la misma región. Sin embargo, puede utilizar un cliente en la región EE. UU. Este (Norte de Virginia) para crear un bucket en cualquier región que se haya lanzado antes del 20 de marzo de 2019. Para obtener más información, consulte Puntos de conexión heredados.

Con estos ejemplos de códigos de los SDK de AWS, se llevan a cabo las siguientes tareas:

- Crear un cliente mediante la especificación explícita de una Región de AWS: en el ejemplo, el cliente utiliza el punto de conexión `s3.us-west-2.amazonaws.com` para comunicarse con Amazon S3. Puede especificar cualquier Región de AWS. Para ver una lista de las Regiones de AWS, consulte Regiones y puntos de enlace en la Referencia general de AWS.
- Enviar una solicitud de creación de bucket mediante la especificación de solo un nombre de bucket: el cliente envía una solicitud a Amazon S3 para crear el bucket en la región donde usted creó un cliente.
- Recuperar la información acerca de la ubicación del bucket: Amazon S3 almacena información de la ubicación del bucket en el subrecurso ubicación asociado con el bucket.

Java

En este ejemplo, se muestra cómo crear un bucket de Amazon S3 con AWS SDK for Java. Para obtener instrucciones sobre cómo crear y probar una muestra funcional, consulte Introducción en la Guía del desarrollador de AWS SDK for Java.

```
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

Para obtener información sobre cómo crear y probar una muestra funcional, consulte la Referencia de la API del SDK de AWS para .NET, versión 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

Para obtener información sobre cómo crear y probar una muestra funcional, consulte el SDK de AWS para Ruby, versión 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__
```

Uso de la AWS CLI

También puede usar AWS Command Line Interface (AWS CLI) para crear un bucket de S3. Para obtener más información, consulte create-bucket en la Referencia de comandos de la AWS CLI.

Para obtener información sobre AWS CLI, consulte ¿Qué es AWS Command Line Interface? en la Guía del usuario de AWS Command Line Interface.

Visualización de las propiedades para un bucket de S3

Puede ver las propiedades de cualquier bucket de Amazon S3 del que sea propietario. Esta configuración incluye lo siguiente:

- Bucket versioning (Control de versiones de bucket): el control de versiones le permite mantener varias versiones de un objeto en un bucket. De forma predeterminada, el control de versiones está deshabilitado para un nuevo bucket. Para obtener más información sobre la habilitación del control de versiones, consulte Habilitar el control de versiones en buckets.
- Tags (Etiquetas): con la asignación de costos de AWS, puede usar etiquetas de bucket para registrar la facturación por el uso de un bucket. Una etiqueta es un par clave-valor que representa una etiqueta que podrá asignar a un bucket. Para obtener más información, consulte Uso de etiquetas de buckets de S3 de asignación de costos.
- Default encryption (Cifrado predeterminado): la habilitación del cifrado predeterminado proporciona cifrado automático del lado del servidor. Amazon S3 cifra un objeto antes de guardarlo en un disco y descifra el objeto al descargarlo. Para obtener más información, consulte Establecer el comportamiento del cifrado predeterminado del lado del servidor para los buckets de Amazon S3.
- Server access logging (Registro de acceso del servidor): el registro de acceso del servidor brinda registros detallados para las solicitudes realizadas a su bucket. De forma predeterminada, Amazon S3 no recopila registros de acceso al servidor. Para obtener información acerca de cómo habilitar el registro de acceso del servidor, consulte Habilitación del registro de acceso al servidor de Amazon S3.
- Eventos de datos de AWS CloudTrail: use CloudTrail para registrar eventos de datos. De forma predeterminada, los registros de seguimiento no registran eventos de datos. Se aplican cargos adicionales a los eventos de datos. Para obtener más información, consulte Registro de eventos de datos para seguimiento en la Guía del usuario de AWS CloudTrail.
- Event notifications (Notificaciones de eventos): puede habilitar ciertos eventos de bucket de Amazon S3 para enviar mensajes de notificación a un destino cuando se producen eventos. Para obtener más información, consulte Habilitación y configuración de notificaciones de eventos mediante la consola de Amazon S3.
- Transfer acceleration (Aceleración de transferencia): permite transferir archivos de forma rápida, fácil y segura entre su cliente y un bucket de S3 a larga distancia. Para obtener información acerca de cómo habilitar Transfer Acceleration, consulte Habilitación y uso de S3 Transfer Acceleration.

- Object Lock (Bloqueo de objetos): use Bloqueo de objetos de S3 para evitar que se elimine o se sobrescriba un objeto durante un periodo de tiempo determinado o de manera indefinida. Para obtener más información, consulte Usar Bloqueo de objetos de S3.
- Requester Pays (Pago por solicitante): puede habilitar el pago por solicitante de modo que el solicitante (en lugar del propietario del bucket) pague las solicitudes y transferencias de datos. Para obtener más información, consulte Utilización de buckets de pago por solicitante para transferencias de almacenamiento y uso.
- Static website hosting (Alojamiento de sitios web estáticos): puede alojar un sitio web estático en Amazon S3. Para obtener más información, consulte Alojamiento de un sitio web estático mediante Amazon S3.

Puede ver las propiedades del bucket mediante los SDK de AWS Management Console, AWS CLI o AWS.

Uso de la consola de S3

1. Inicie sesión en la AWS Management Console y abra la consola de Amazon S3 en https://console.aws.amazon.com/s3/.
2. En la lista Buckets, seleccione el nombre del bucket cuyas propiedades desea ver.
3. Elija la pestaña Propiedades.
4. En la página Propiedades, puede configurar las anteriores propiedades para el bucket.

Uso de la AWS CLI

Visualización de las propiedades del bucket con la AWS CLI

Los siguientes comandos muestran cómo puede utilizar la AWS CLI para mostrar diferentes propiedades de bucket.

En el siguiente ejemplo se devuelve el conjunto de etiquetas asociado al bucket *amzn-s3-demo-bucket1*. Para obtener más información acerca de las etiquetas de buckets, consulte Uso de etiquetas de buckets de S3 de asignación de costos.

```
aws s3api get-bucket-tagging --bucket amzn-s3-demo-bucket1
```

Para obtener más información y ejemplos, consulte get-bucket-tagging en la referencia de comandos de AWS CLI.

En el siguiente ejemplo se devuelve el estado de control de versiones del bucket *amzn-s3-demo-bucket1*. Para obtener más información sobre el control de versiones de buckets, consulte Usar el control de versiones en buckets de S3.

```
aws s3api get-bucket-versioning --bucket amzn-s3-demo-bucket1
```

Para obtener más información y ejemplos, consulte get-bucket-versioning en la referencia de comandos de AWS CLI.

En el siguiente ejemplo se devuelve la configuración de cifrado predeterminada del bucket *amzn-s3-demo-bucket1*. De forma predeterminada, todos los buckets tienen una configuración de cifrado predeterminada que utiliza el cifrado del servidor con claves administradas de Amazon S3 (SSE-S3). Para obtener información sobre el cifrado predeterminado de buckets, consulte Establecer el comportamiento del cifrado predeterminado del lado del servidor para los buckets de Amazon S3.

```
aws s3api get-bucket-encryption --bucket amzn-s3-demo-bucket1
```

Para obtener más información y ejemplos, consulte get-bucket-encryption en la referencia de comandos de AWS CLI.

En el siguiente ejemplo se devuelve la configuración de notificación del bucket *amzn-s3-demo-bucket1*. Para obtener información acerca de las notificaciones de eventos de buckets, consulte Notificaciones de eventos de Amazon S3.

```
aws s3api get-bucket-notification-configuration --bucket amzn-s3-demo-bucket1
```

Para obtener más información y ejemplos, consulte get-bucket-notification-configuration en la referencia de comandos de AWS CLI.

En el siguiente ejemplo se devuelve el estado de registro del bucket *amzn-s3-demo-bucket1*. Para obtener información acerca del registro de buckets, consulte Registro de solicitudes con registro de acceso al servidor.

```
aws s3api get-bucket-logging --bucket amzn-s3-demo-bucket1
```

Para obtener más información y ejemplos, consulte get-bucket-logging en la referencia de comandos de AWS CLI.

Uso de los AWS SDK

Para ver ejemplos de cómo devolver las propiedades del bucket con los AWS SDK, como el control de versiones, las etiquetas y mucho más, consulte Acciones de Amazon S3 con SDK de AWS.

Para obtener información general sobre el uso de diferentes SDK de AWS, consulte Desarrollo con Amazon S3 mediante los SDK de AWS.

Vaciar un bucket

Puede vaciar el contenido de un bucket mediante la consola de Amazon S3, los SDK de AWS o AWS Command Line Interface (AWS CLI). Cuando vacía un bucket, elimina todos los objetos, pero conserva el bucket. Una vez que lo hace, no se puede deshacer la acción. Es posible que se eliminen los objetos agregados al bucket mientras la acción de vaciado del bucket esté en curso. Todos los objetos (incluidas todas las versiones de objetos y los marcadores de eliminación) del bucket se deben eliminar antes de poder eliminar el propio bucket.

Cuando se vacía un bucket que tiene habilitado o suspendido el control de versiones de S3, todas las versiones de todos los objetos del bucket se eliminarán. Para obtener más información, consulte Trabajar con objetos en un bucket con control de versiones habilitado.

También puede especificar una configuración del ciclo de vida de un bucket para que provoque el vencimiento de objetos de modo que Amazon S3 pueda eliminarlos. Para obtener más información, consulte Configuración de un ciclo de vida en un bucket. Para vaciar un bucket de gran tamaño, le recomendamos que utilice una regla de configuración del ciclo de vida de S3. La caducidad del ciclo de vida es un proceso asincrónico, por lo que la regla puede tardar algunos días en ejecutarse antes de que el bucket se quede vacío. Tras la primera vez que Amazon S3 ejecuta la regla, todos los objetos que cumplen los requisitos de caducidad se marcan para su eliminación. Se le dejará de cobrar por los objetos que estén marcados para ser eliminados. Para obtener más información, consulte How do I empty an Amazon S3 bucket using a lifecycle configuration rule? (¿Cómo puedo vaciar un bucket de Amazon S3 mediante una regla de configuración del ciclo de vida?).

Uso de la consola de S3

Puede utilizar la consola de Amazon S3 para vaciar un bucket, lo que elimina todos los objetos del bucket sin eliminar el bucket.

Para vaciar un bucket de S3

1. Inicie sesión en la AWS Management Console y abra la consola de Amazon S3 en https://console.aws.amazon.com/s3/.
2. En la lista Bucket name (Nombre del bucket), seleccione la opción junto al nombre del bucket que desea vaciar y, a continuación, elija Empty (Vaciar).
3. En la página Empty bucket (Vaciar bucket), confirme que desea vaciar el bucket introduciendo el nombre del bucket en el campo de texto y, a continuación, elija Empty (Vaciar).
4. Supervise el progreso del proceso de vaciado del bucket en la página Empty bucket: Status (Vaciado del bucket: estado).

Mediante AWS CLI

Puede vaciar un bucket con la AWS CLI solo si el bucket no tiene habilitado el control de versiones. Si el control de versiones no está habilitado, puede usar el comando `rm` (eliminar) de AWS CLI con el parámetro `--recursive` para vaciar un bucket (o eliminar un subconjunto de objetos con un prefijo de nombre de clave específico).

El siguiente comando `rm` elimina objetos con el prefijo de nombre de clave `doc`, por ejemplo, `doc/doc1` y `doc/doc2`.

```
$ aws s3 rm s3://bucket-name/doc --recursive
```

Use el siguiente comando para eliminar todos los objetos sin especificar ningún prefijo.

```
$ aws s3 rm s3://bucket-name --recursive
```

Para obtener más información, consulte Uso de comandos de S3 de alto nivel con la AWS CLI en la Guía del usuario de AWS Command Line Interface.

> **Note**
>
> No puede eliminar objetos de un bucket que tenga habilitado el control de versiones. Amazon S3 agregará un marcador de eliminación al eliminar un objeto, que es lo que hace este comando. Para obtener más información sobre el control de versiones del bucket de S3, consulte Usar el control de versiones en buckets de S3.

Uso de la SDKs AWS

Puede usar los SDK de AWS para vaciar un bucket o eliminar un subconjunto de objetos con un prefijo de nombre de clave específico.

Si desea ver un ejemplo de cómo vaciar un bucket utilizando AWS SDK for Java, consulte Eliminar un bucket. Con este código eliminará todos los objetos, independientemente de si el bucket tiene habilitado el control de versiones, y después se eliminará el bucket. Si quiere limitarse a vaciar el bucket, asegúrese de eliminar la instrucción que provoca la eliminación del bucket.

Para obtener más información acerca del uso de otros SDK de AWS, consulte Herramientas para Amazon Web Services.

Uso de una configuración del ciclo de vida

Para vaciar un bucket de gran tamaño, le recomendamos que utilice una regla de configuración del ciclo de vida de S3. La caducidad del ciclo de vida es un proceso asincrónico, por lo que la regla puede tardar algunos días en ejecutarse antes de que el bucket se quede vacío. Tras la primera vez que Amazon S3 ejecuta la regla, todos los objetos que cumplen los requisitos de caducidad se marcan para su eliminación. Se le dejará de cobrar por los objetos que estén marcados para ser eliminados. Para obtener más información, consulte How do I empty an Amazon S3 bucket using a lifecycle configuration rule? (¿Cómo puedo vaciar un bucket de Amazon S3 mediante una regla de configuración del ciclo de vida?).

Si utiliza una configuración del ciclo de vida para vaciar el bucket, la configuración debería incluir versiones actuales, versiones no actuales, marcadores de eliminación y cargas multiparte incompletas.

Puede agregar reglas de configuración del ciclo de vida para provocar el vencimiento de todos o de un subconjunto de los objetos que tenga un prefijo de nombre de clave específico. Por ejemplo, para eliminar todos los objetos de un bucket, puede configurar una regla del ciclo de vida que haga que venzan los objetos un día después de su creación.

Amazon S3 admite una regla de ciclo de vida del bucket que puede utilizar para detener las cargas multiparte que no se completan dentro de un número especificado de días después de iniciarse. Le recomendamos que configure esta regla de ciclo de vida para minimizar los costos de almacenamiento. Para obtener más información, consulte Configuración de una política de ciclo de vida del bucket para eliminar cargas multiparte incompletas.

Para obtener más información acerca del uso de una configuración de ciclo de vida para vaciar un bucket, consulte Configuración de un ciclo de vida en un bucket y Vencimiento de objetos.

Vaciar un bucket con AWS CloudTrail configurado

AWS CloudTrail rastrea los eventos de datos en el nivel de objeto en un bucket de Amazon S3, como la eliminación de objetos. Si utiliza un bucket como destino para registrar los eventos de CloudTrail y está eliminando objetos de ese mismo bucket, es posible que esté creando nuevos objetos mientras vacía el bucket. Para evitarlo, detenga los registros de seguimiento AWS CloudTrail. Para obtener más información sobre cómo detener los registros de seguimiento de CloudTrail de los eventos de registro, consulte Desactivar el registro de un registro de seguimiento en la Guía del usuario de AWS CloudTrail.

Otra alternativa para evitar que los registros de seguimiento de CloudTrail se agreguen al bucket es agregar una instrucción de denegación `s3:PutObject` a la política del bucket. Si desea almacenar objetos nuevos en el bucket más adelante, tendrá que eliminar esta instrucción `s3:PutObject` de denegación. Para obtener más información, consulte Operaciones con objetos y Elementos de la política JSON de IAM: Efecto en la Guía del usuario de IAM.

Eliminar un bucket

Puede eliminar un bucket vacío de Amazon S3. Antes de eliminar un bucket, tenga en cuenta lo siguiente:

- Los nombres de bucket son únicos. Si elimina un bucket, otro usuario de AWS podrá utilizar el nombre.
- Si el bucket aloja un sitio web estático y ha creado y configurado una zona alojada de Amazon Route 53 como se describe en Tutorial: Configuración de un sitio web estático mediante un dominio personalizado registrado con Route 53, debe limpiar la configuración de la zona alojada de Route 53 que está relacionada con el bucket. Para obtener más información, consulte Paso 2: eliminar la zona alojada en Route 53.
- Si el bucket recibe datos de registro de Elastic Load Balancing (ELB), recomendamos que detenga el envío de registros de ELB al bucket antes de eliminarlo. Después de eliminar el bucket, si otro usuario crea un bucket con el mismo nombre, existe la posibilidad de que sus datos se registro se envíen a ese bucket. Para obtener información acerca de los registros de acceso de ELB, consulte Registros de acceso en la Guía del usuario para balanceadores de carga clásicos y Registros de acceso en la Guía del usuario para balanceadores de carga de aplicaciones.

Resolución de problemas

Si no puede eliminar un bucket de Amazon S3, tenga en cuenta lo siguiente:

- Asegúrese de que el bucket esté vacío: solo se pueden eliminar buckets que no tengan ningún objeto en ellos. Asegúrate de que el bucket esté vacío.
- Asegúrese de que no haya ningún punto de acceso conectado: solo se pueden eliminar buckets que no tengan ningún punto de acceso conectado a ellos. Elimine todos los puntos de acceso que estén conectados al bucket antes de eliminarlo.
- Políticas de control de servicio (SCP) de AWS Organizations: una política de control de servicio puede denegar el permiso de eliminación de un bucket. Para obtener información acerca de SCP, consulte Políticas de control de servicios en la Guía del usuario de AWS Organizations.
- S3: permisos DeleteBucket: si no puede eliminar un bucket, trabaje con el administrador de IAM para confirmar que tiene permisos de `s3:DeleteBucket`. Para obtener información acerca de cómo consultar o actualizar permisos de IAM, consulte Cambio de permisos de un usuario de IAM en la Guía del usuario de IAM.
- s3: afirmación de denegación de DeleteBucket: si tiene permisos de `s3:DeleteBucket` en su política de IAM y no puede eliminar un bucket, la política del bucket podría incluir una afirmación de denegación `s3:DeleteBucket`. Los buckets creados por ElasticBeanStalk tienen una política que contiene esta instrucción de forma predeterminada. Antes de poder eliminar el bucket, debe eliminar esta instrucción o la política del bucket.

> **Important**
>
> Los nombres de bucket son únicos. Si elimina un bucket, otro usuario de AWS podrá utilizar el nombre. Si desea seguir utilizando el mismo nombre de bucket, no elimine el bucket. Le recomendamos que vacíe el bucket y lo conserve.

Uso de la consola de S3

Para eliminar un bucket de S3

1. Inicie sesión en la AWS Management Console y abra la consola de Amazon S3 en https://console.aws.amazon.com/s3/.
2. En la lista Buckets (Buckets), seleccione la opción situada junto al nombre del bucket que desea eliminar y, a continuación, elija Delete (Eliminar) en la parte superior de la página.

3. En la página Delete bucket (Eliminar bucket) confirme que desea eliminar el bucket introduciendo el nombre del bucket en el campo de texto y, a continuación, elija Delete bucket (Eliminar bucket).

> **Note**
>
> Si el bucket contiene objetos, vacíelo antes de eliminarlo seleccionando el vínculo empty bucket configuration (vaciar configuración de bucket) en la alerta de error This bucket is not empty (Este bucket no está vacío) y siguiendo las instrucciones de la página Empty bucket (Vaciar bucket). A continuación, vuelva a la página Delete bucket (Eliminar bucket) y elimine el bucket.

4. Para verificar que ha eliminado el bucket, abra la lista Buckets e ingrese el nombre del bucket que ha eliminado. Si no encuentra el bucket, la eliminación se ha realizado correctamente.

Uso de AWS SDK para Java

En el siguiente ejemplo, se muestra cómo eliminar un bucket con el SDK de AWS para Java. En primer lugar, el código elimina los objetos del bucket, y a continuación elimina el bucket. Para obtener más información sobre otros SDK de AWS, consulte Herramientas para Amazon Web Services.

Java

En el siguiente ejemplo de Java se elimina un bucket que contiene objetos. En el ejemplo se eliminan todos los objetos y, a continuación, se elimina el bucket. El ejemplo funciona también con buckets con o sin control de versiones habilitado.

> **Note**
>
> En buckets sin control de versiones habilitado, puede eliminar todos los objetos directamente y después eliminar el bucket. En buckets con control de versiones habilitado, debe eliminar todas las versiones de objetos antes de eliminar el bucket.

Para obtener instrucciones sobre la creación y comprobación de una muestra funcional, consulte Introducción en la Guía del desarrollador de AWS SDK for Java.

```
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();
        }
```

```
    }
}
```

Uso de la AWS CLI

Puede eliminar un bucket que contenga objetos con la AWS CLI si no tiene habilitado el control de versiones. Si elimina un bucket que contenga objetos, se eliminarán todos los objetos del bucket de forma permanente, incluidos los objetos que pasaron a la clase de almacenamiento `S3 Glacier`.

Si el bucket no tiene habilitado el control de versiones, puede utilizar el comando de la AWS CLI `rb` (eliminar bucket) con el parámetro `--force` para eliminar el bucket y todos los objetos que contiene. Este comando elimina todos los objetos en primer lugar, y después elimina el bucket.

Si el control de versiones está habilitado, los objetos con control de versiones no se eliminarán en este proceso, lo que provocaría un error en la eliminación del bucket porque el bucket no estaría vacío. Para obtener más información sobre cómo eliminar objetos con control de versiones, consulte Eliminación de versiones de objetos.

```
$ aws s3 rb s3://bucket-name --force
```

Para obtener más información, consulte Uso de comandos de S3 de alto nivel con la AWS Command Line Interface en la Guía del usuario de AWS Command Line Interface.

Establecer el comportamiento del cifrado predeterminado del lado del servidor para los buckets de Amazon S3

> **Important**
>
> Amazon S3 aplica ahora el cifrado del servidor con claves administradas por Amazon S3 (SSE-S3) como el nivel básico de cifrado para cada bucket de Amazon S3. Desde el 5 de enero de 2023, todas las cargas de objetos nuevos a Amazon S3 se cifran automáticamente sin costo adicional y sin afectar al rendimiento. El estado de cifrado automático para la configuración de cifrado predeterminada en el bucket de S3 y para cargas de objetos nuevos está disponible en registros de AWS CloudTrail, Inventario de S3, Lente de almacenamiento de S3, la consola de Amazon S3 y como encabezado de respuesta a la API de Amazon

S3 adicional en AWS Command Line Interface y los SDK de AWS. Para obtener más información, consulte Preguntas frecuentes del cifrado predeterminado.

Todos los bucket de Amazon S3 tienen el cifrado configurado de forma predeterminada y los objetos se cifran de forma automática con el cifrado del servidor mediante claves administradas de Amazon S3 (SSE-S3). Esta configuración de cifrado se aplica a todos los objetos de sus buckets de Amazon S3.

Si necesita más control sobre sus claves, por ejemplo, administrar la rotación de claves y las concesiones de las políticas de acceso, puede optar por utilizar el cifrado del servidor con claves de AWS Key Management Service (AWS KMS) (SSE-KMS) o con el cifrado del servidor de doble capa con claves de AWS KMS (DSSE-KMS). Para obtener más información sobre las claves de KMS, consulte Edición de claves en la Guía para desarrolladores de AWS Key Management Service.

Note

Hemos cambiado los buckets para cifrar automáticamente las cargas de objetos nuevos. Si anteriormente creó un bucket sin cifrado predeterminado, Amazon S3 habilitará el cifrado de forma predeterminada para el bucket mediante SSE-S3. No se modificará la configuración de cifrado predeterminada para un bucket existente que ya tenga configurado SSE-S3 o SSE-KMS. Si desea cifrar sus objetos con SSE-KMS, debe cambiar el tipo de cifrado en la configuración del bucket. Para obtener más información, consulte Uso del cifrado del servidor con claves de AWS KMS (SSE-KMS).

Cuando configura el bucket para que utilice el cifrado predeterminado con SSE-KMS, también puede habilitar las claves de bucket de S3 para reducir el tráfico de Amazon S3 a AWS KMS y el costo del cifrado. Para obtener más información, consulte Reducción del costo de SSE-KMS con las claves de bucket de Amazon S3.

Para identificar los buckets que tienen el SSE-KMS habilitado para el cifrado predeterminado, puede utilizar las métricas de la Lente de almacenamiento de Amazon S3. Lente de almacenamiento de S3 es una función de análisis de almacenamiento en la nube que puede utilizar para obtener visibilidad en toda la organización sobre el uso y la actividad del almacenamiento de objetos. Para obtener más información, consulte Uso de Lente de almacenamiento de S3 para proteger los datos.

Si usa el cifrado del servidor, Amazon S3 cifra un objeto antes de guardarlo en el disco y lo descifra al descargar el objeto. Para obtener más información sobre cómo proteger los datos mediante el

cifrado del servidor y la administración de claves de cifrado, consulte Protección de los datos con el cifrado del servidor.

Para obtener más información acerca de los permisos necesarios para el cifrado predeterminado, consulte PutBucketEncryption en la Referencia de API de Amazon Simple Storage Service.

Puede configurar el cifrado predeterminado de Amazon S3 para un bucket de S3 a través de la consola de Amazon S3, los SDK de AWS, la API de REST de Amazon S3 y la AWSCommand Line Interface (AWS CLI CLI).

Cifrado de objetos existentes

Para cifrar los objetos de Amazon S3 no cifrados existentes, puede utilizar las operaciones por lotes de Amazon S3. A las operaciones por lotes de S3 se les proporciona una lista de objetos en los que deben actuar. Las operaciones por lotes llaman a la API correspondiente para llevar a cabo la operación especificada. Puede utilizar la operación de copia de la herramienta de operaciones por lotes para copiar objetos existentes sin cifrar y escribirlos como objetos cifrados en el mismo bucket. Un solo trabajo de la herramienta de operaciones por lotes puede realizar la operación especificada en miles de millones de objetos. Para obtener más información, consulte Realización de operaciones por lotes a gran escala en objetos de Amazon S3 y la publicación Encrypting objects with Amazon S3 Batch Operations del Blog de almacenamiento de AWS.

También puede cifrar los objetos existentes mediante la operación de la API `CopyObject` o el comando `copy-object` AWS CLI. Para obtener más información, consulte la publicación Encrypting existing Amazon S3 objects with the AWS CLI del Blog de almacenamiento de AWS.

> **Note**
>
> Los buckets de Amazon S3 con cifrado de bucket predeterminado con SSE-KMS no se pueden utilizar como buckets de destino para the section called “Registro de acceso al servidor”. Solo se admite el cifrado predeterminado SSE-S3 para los buckets de destino del registro de acceso al servidor.

Uso del cifrado SSE-KMS para operaciones entre cuentas

Tenga en cuenta lo siguiente cuando utilice el cifrado para operaciones entre cuentas:

- Si no se proporciona un nombre de recurso de Amazon (ARN) o un alias de AWS KMS key en el momento de la solicitud, ni a través de la configuración de cifrado predeterminado del bucket, se usa la Clave administrada de AWS (`aws/s3`).
- Si está cargando o accediendo a objetos de S3 usando las entidades principales de AWS Identity and Access Management (IAM) que están en la misma Cuenta de AWS que la clave de KMS, puede usar la Clave administrada de AWS (`aws/s3`).
- Use una clave administrada por el cliente si desea conceder acceso entre cuentas a sus objetos de S3. Puede configurar la política de una clave administrada por el cliente para permitir el acceso desde otra cuenta.
- Si especifica una clave de KMS administrada por el cliente, le recomendamos que use un ARN totalmente cualificado de la clave de KMS. Si, en su lugar, utiliza un alias de clave de KMS, AWS KMS resolverá la clave dentro de la cuenta del solicitante. Esto puede dar como resultado datos cifrados con una clave de KMS que pertenece al solicitante y no al propietario del bucket.
- Debe especificar una clave para la que el solicitante le haya concedido permiso `Encrypt`. Para obtener más información, consulte Permitir a los usuarios de claves utilizar una clave de KMS para las operaciones criptográficas en la Guía para desarrolladores de AWS Key Management Service.

Para obtener más información acerca de cuándo utilizar claves administradas por el cliente y las claves de KMS administradas por AWS, consulte ¿Debo usar una clave administrada por Clave administrada de AWS o una clave administrada por el cliente para cifrar mis objetos en Amazon S3?

Uso del cifrado predeterminado con la replicación

Cuando habilita el cifrado predeterminado para un bucket de destino de replicación, se aplica el siguiente comportamiento de cifrado:

- Si los objetos del bucket de origen no están cifrados, los objetos de réplica del bucket de destino se cifran mediante la configuración de cifrado predeterminado del bucket de destino. Como resultado, las etiquetas de entidad (ETags) de los objetos de origen difieren de las ETags de los objetos de réplica. Si tiene aplicaciones que utilizan ETags, deberá actualizarlas para tener en cuenta esta diferencia.
- Si los objetos del bucket de origen se cifran mediante el cifrado del servidor con claves administradas por Amazon S3 (SSE-S3), el cifrado del servidor con claves de AWS Key Management Service (AWS KMS) (SSE-KMS) o con cifrado del servidor de doble capa con claves de AWS KMS (DSSE-KMS), los objetos de réplica del bucket de destino utilizarán el mismo tipo

de cifrado que los objetos de origen. La configuración de cifrado predeterminado del bucket de destino no se utiliza.

Para obtener más información acerca del uso del cifrado predeterminado con SSE-KMS, consulte Replicar objetos cifrados.

Uso de claves de bucket de Amazon S3 con cifrado predeterminado

Si configura el bucket para que utilice el cifrado predeterminado para SSE-KMS en los objetos nuevos, también puede configurar las claves de bucket de S3. Las claves de bucket de S3 reducen el número de transacciones de Amazon S3 a AWS KMS para rebajar el costo de SSE-KMS.

Cuando configura el bucket a fin de que utilice claves de bucket de S3 para SSE-KMS en objetos nuevos, AWS KMS genera una clave de bucket que se utiliza con el fin de crear una clave de datos única para los objetos del bucket. Esta clave de bucket de S3 se utiliza durante un periodo limitado dentro de Amazon S3, lo que reduce la necesidad de que Amazon S3 realice solicitudes a AWS KMS para completar las operaciones de cifrado.

Para obtener más información sobre el uso de claves de bucket de S3, consulte Uso de claves de bucket de Amazon S3.

Configuración del cifrado predeterminado

> **Important**
>
> Amazon S3 aplica ahora el cifrado del servidor con claves administradas por Amazon S3 (SSE-S3) como el nivel básico de cifrado para cada bucket de Amazon S3. Desde el 5 de enero de 2023, todas las cargas de objetos nuevos a Amazon S3 se cifran automáticamente sin costo adicional y sin afectar al rendimiento. El estado de cifrado automático para la configuración de cifrado predeterminada en el bucket de S3 y para cargas de objetos nuevos está disponible en registros de AWS CloudTrail, Inventario de S3, Lente de almacenamiento de S3, la consola de Amazon S3 y como encabezado de respuesta a la API de Amazon S3 adicional en AWS Command Line Interface y los SDK de AWS. Para obtener más información, consulte Preguntas frecuentes del cifrado predeterminado.

Los bucket de Amazon S3 tienen el cifrado de buckets activado de forma predeterminada y los objetos nuevos se cifran automáticamente mediante el cifrado del servidor con claves administradas

de Amazon S3 (SSE-S3). Este cifrado se aplica a todos los objetos nuevos de sus buckets de Amazon S3 y no tiene ningún costo para usted.

Si necesita más control sobre las claves de cifrado, por ejemplo, administrar la rotación de claves y las concesiones de las políticas de acceso, puede optar por utilizar el cifrado del servidor con claves de AWS Key Management Service (AWS KMS) (SSE-KMS) o el cifrado del servidor de doble capa con claves de AWS KMS (DSSE-KMS). Para obtener más información sobre SSE-KMS, consulte Especificación del cifrado del lado del servidor con AWS KMS (SSE-KMS). Para obtener más información sobre DSSE-KMS, consulte the section called “Cifrado del servidor de doble capa (SSE-KMS)”.

Si desea utilizar una clave de KMS propiedad de una cuenta diferente, primero debe tener permiso para utilizar la clave. Para obtener más información sobre los permisos entre cuentas para las claves de KMS, consulte Crear claves de KMS que otras cuentas puedan utilizar en la Guía para desarrolladores de AWS Key Management Service.

Al establecer el cifrado del bucket predeterminado en SSE-KMS, también puede configurar una clave de bucket de S3 para reducir los costos de las solicitudes de AWS KMS. Para obtener más información, consulte Reducción del costo de SSE-KMS con las claves de bucket de Amazon S3.

> **Note**
>
> Si utiliza PutBucketEncryption para establecer a SSE-KMS el cifrado predeterminado de su bucket, deberá verificar que el ID de su clave KMS es correcto. Amazon S3 no valida el ID de clave KMS proporcionado en las solicitudes de PutBucketEncryption.

No se aplican cargos adicionales por usar el cifrado predeterminado de buckets de S3. Las solicitudes para configurar el comportamiento de cifrado predeterminado generan cargos por solicitudes de Amazon S3 estándar. Para obtener información acerca de los precios, consulte Precios de Amazon S3. Para SSE-KMS y DSSE-KMS, se aplican los cargos de AWS KMS que se muestran en Precios de AWS KMS.

No se admite el cifrado predeterminado del servidor con claves proporcionadas por el cliente (SSE-C).

Puede configurar el cifrado predeterminado de Amazon S3 para un bucket de S3 a través de la consola de Amazon S3, los SDK de AWS, la API de REST de Amazon S3 y la AWS Command Line Interface (AWS CLI).

Cambios para tener en cuenta antes de habilitar el cifrado predeterminado

Después de habilitar el cifrado predeterminado para un bucket, se aplica el siguiente comportamiento de cifrado:

- No hay ninguna variación en el cifrado de los objetos que existían en el bucket antes de que se habilitara el cifrado predeterminado.
- Cuando carga objetos después de habilitar el cifrado predeterminado:
  - Si los encabezados de las solicitudes PUT no incluyen información de cifrado, Amazon S3 utiliza la configuración de cifrado predeterminada del bucket para cifrar los objetos.
  - Si los encabezados de las solicitudes PUT incluyen información de cifrado, Amazon S3 utiliza la información de cifrado de la solicitud PUT para cifrar los objetos antes de guardarlos en Amazon S3.
- Si utiliza la opción de SSE-KMS o DSSE-KMS para la configuración de cifrado predeterminado, se le aplicarán las cuotas de solicitudes por segundo (RPS) de AWS KMS. Para obtener más información acerca de las cuotas de AWS KMS y cómo solicitar un aumento de cuota, consulte Cuotas en la Guía para desarrolladores de AWS Key Management Service.

> **Note**
>
> Los objetos cargados antes de que se habilitara el cifrado predeterminado no se cifrarán. Para obtener más información sobre el cifrado de objetos existentes, consulte the section called “Establecer el cifrado predeterminado de un bucket”.

Uso de la consola de S3

Para configurar el cifrado predeterminado en un bucket de Amazon S3

1. Inicie sesión AWS Management Console Management Console y abra la consola de Amazon S3 en https://console.aws.amazon.com/s3/.
2. En el panel de navegación izquierdo, elija Instancias.
3. En la lista Buckets, elija el nombre del bucket en cuestión.
4. Elija la pestaña Propiedades.
5. En Cifrado predeterminado, elija Editar.
6. Para configurar el cifrado, elija una de las siguientes opciones en Tipo de cifrado:

- Cifrado del servidor con claves administradas por Amazon S3 (SSE-S3)
- Cifrado del servidor con claves de AWS Key Management Service (SSE-KMS)
- Cifrado del servidor de doble capa con claves de AWS Key Management Service (DSSE-KMS)

> **Important**
>
> Si utiliza las opciones SSE-KMS o DSSE-KMS para la configuración del cifrado predeterminado, se le aplicarán las cuotas de solicitudes por segundo (RPS) de AWS KMS. Para obtener más información acerca de las cuotas de AWS KMS y cómo solicitar un aumento de cuota, consulte Cuotas en la Guía para desarrolladores de AWS Key Management Service.

Los buckets y los objetos nuevos se cifran de forma predeterminada con SSE-S3, a menos que especifique otro tipo de cifrado predeterminado para sus buckets. Para obtener más información acerca del cifrado predeterminado, consulte Establecer el comportamiento del cifrado predeterminado del lado del servidor para los buckets de Amazon S3.

Para obtener más información sobre el uso del cifrado del lado del servidor de Amazon S3 para cifrar los datos, consulte Uso del cifrado del servidor con claves administradas por Amazon S3 (SSE-S3).

7. Si elige el Cifrado del servidor con claves de AWS Key Management Service (SSE-KMS) o el Cifrado del servidor de doble capa con claves de AWS Key Management Service (DSSE-KMS), haga lo siguiente:

   a. En Clave de AWS KMS, especifique su clave de KMS de una de las siguientes maneras:

      - Para seleccionar de una lista de claves de KMS disponibles, marque Elija entre sus claves de AWS KMS keys y seleccione su clave de KMS de la lista de claves disponibles.

        En esta lista aparecen tanto la Clave administrada de AWS (`aws/s3`) como las claves administradas por el cliente. Para obtener más información acerca de las claves administradas por el cliente, consulte Claves de cliente y claves de AWS en la Guía para desarrolladores de AWS Key Management Service.

      - Para introducir el ARN de la clave de KMS, elija Introducir el ARN de la AWS KMS key e introduzca el ARN de la clave de KMS en el campo que aparece.

- Para crear una nueva clave administrada por el cliente en la consola de AWS KMS, elija Crear una clave de KMS.

  Para obtener más información acerca de cómo crear una AWS KMS key, consulte Creación de claves en la AWS Key Management Service Guía para desarrolladores.

> **Important**
>
> Solo puede utilizar las claves de KMS que estén habilitadas en la misma Región de AWS que el bucket. Cuando elige Choose from your KMS master keys (Elegir entre las claves raíz de KMS), la consola de S3 solo muestra 100 claves de KMS por región. Si tiene más de 100 claves de KMS en la misma región, solo puede ver las primeras 100 claves de KMS en la consola S3. Para utilizar una clave de KMS que no aparezca en la consola, elija Introducir el ARN de AWS KMS key y escriba el ARN de la clave de KMS.
> Cuando utilice una AWS KMS key para el cifrado en el lado del servidor en Amazon S3, debe elegir una clave de cifrado de KMS simétrica. Amazon S3 solo admite claves KMS de cifrado simétricas. Para obtener más información sobre estas claves, consulte Symmetric encryption KMS keys (Claves de KMS de cifrado simétricas) en la Guía para desarrolladores de AWS Key Management Service.

Para obtener más información sobre el uso de SSE-KMS con Amazon S3, consulte Uso del cifrado del servidor con claves de AWS KMS (SSE-KMS). Para obtener más información sobre el uso de DSSE-KMS, consulte the section called “Cifrado del servidor de doble capa (SSE-KMS)”.

b. Si configura el bucket para que use el cifrado predeterminado con SSE-KMS, también puede habilitar una clave de bucket de S3. Las claves de bucket de S3 reducen el costo del cifrado al reducir el tráfico de solicitudes de Amazon S3 a AWS KMS. Para obtener más información, consulte Reducción del costo de SSE-KMS con las claves de bucket de Amazon S3.

   Para utilizar las claves de bucket de S3, en Clave de bucket, seleccione Habilitar.

> **Note**
>
> Las claves de bucket de S3 no son compatibles con DSSE-KMS.

8. Elija Guardar cambios.

Uso de la AWS CLI

En estos ejemplos se muestra cómo configurar el cifrado predeterminado con SSE-S3 o SSE-KMS con una clave de bucket de S3.

Para obtener más información acerca del cifrado predeterminado, consulte Establecer el comportamiento del cifrado predeterminado del lado del servidor para los buckets de Amazon S3. Para obtener más información acerca del uso de la AWS CLI para configurar el cifrado predeterminado, consulte put-bucket-encryption.

Example — Cifrado predeterminado con SSE-S3

En este ejemplo se configura el cifrado de bucket predeterminado con las claves administradas de Amazon S3.

```
aws s3api put-bucket-encryption --bucket amzn-s3-demo-bucket --server-side-encryption-
configuration '{
    "Rules": [
        {
            "ApplyServerSideEncryptionByDefault": {
                "SSEAlgorithm": "AES256"
            }
        }
    ]
}'
```

Example — Cifrado predeterminado con SSE-KMS usando una clave de bucket de S3

En este ejemplo se configura el cifrado de bucket predeterminado con SSE-KMS mediante una clave de bucket de S3.

```
aws s3api put-bucket-encryption --bucket amzn-s3-demo-bucket --server-side-encryption-
configuration '{
    "Rules": [
```

```
            {
                "ApplyServerSideEncryptionByDefault": {
                    "SSEAlgorithm": "aws:kms",
                    "KMSMasterKeyID": "KMS-Key-ARN"
                },
                "BucketKeyEnabled": true
            }
        ]
    }'
```

Uso de la API de REST

Utilice la operación de la API de REST `PutBucketEncryption` para habilitar el cifrado predeterminado y establecer el tipo de cifrado del servidor para usar SSE-S3, SSE-KMS o DSSE-KMS.

Para obtener más información, consulte PutBucketEncryption en la Referencia de la API de Amazon Simple Storage Service.

Monitoreo del cifrado predeterminado con AWS CloudTrail y Amazon EventBridge

> **Important**
>
> Amazon S3 aplica ahora el cifrado del servidor con claves administradas por Amazon S3 (SSE-S3) como el nivel básico de cifrado para cada bucket de Amazon S3. Desde el 5 de enero de 2023, todas las cargas de objetos nuevos a Amazon S3 se cifran automáticamente sin costo adicional y sin afectar al rendimiento. El estado de cifrado automático para la configuración de cifrado predeterminada en el bucket de S3 y para cargas de objetos nuevos está disponible en registros de AWS CloudTrail, Inventario de S3, Lente de almacenamiento de S3, la consola de Amazon S3 y como encabezado de respuesta a la API de Amazon S3 adicional en AWS Command Line Interface y los SDK de AWS. Para obtener más información, consulte Preguntas frecuentes del cifrado predeterminado.

Puede realizar un seguimiento de las solicitudes de configuración de cifrado predeterminado para los buckets de Amazon S3 mediante eventos de AWS CloudTrail. Los siguientes nombres de eventos de API se utilizan en los registros de CloudTrail:

- `PutBucketEncryption`

- `GetBucketEncryption`
- `DeleteBucketEncryption`

También puede crear reglas de EventBridge para que coincidan con los eventos de CloudTrail para estas llamadas a la API. Para obtener más información acerca de los eventos de CloudTrail, consulte Habilitar el registro de objetos en un bucket mediante la consola. Para obtener más información acerca de los eventos de EventBridge, consulte Eventos de Servicios de AWS.

Puede utilizar registros de CloudTrail para acciones de Amazon S3 de objeto y así realizar un seguimiento de las solicitudes `PUT` y `POST` a Amazon S3. Puede utilizar estas acciones para verificar si se usa el cifrado predeterminado para cifrar los objetos cuando las solicitudes `PUT` entrantes no tienen encabezados de cifrado.

Cuando Amazon S3 cifra un objeto utilizando la configuración de cifrado predeterminada, el registro incluye los siguientes campos como el par de nombre-valor: `"SSEApplied":"Default_SSE_S3"`, `"SSEApplied":"Default_SSE_KMS"` o `"SSEApplied":"Default_DSSE_KMS"`.

Cuando Amazon S3 cifra un objeto utilizando los encabezados de cifrado `PUT`, el registro incluye uno de los siguientes campos como el par de nombre-valor: `"SSEApplied":"SSE_S3"`, `"SSEApplied":"SSE_KMS"`, `"SSEApplied":"DSSE_KMS"` o `"SSEApplied":"SSE_C"`.

Para las cargas multiparte, esta información se incluye en las solicitudes de la operación de la API `InitiateMultipartUpload`. Para obtener más información sobre el uso de CloudTrail y CloudWatch, consulte Monitorización de Amazon S3.

Uso de Mountpoint para Amazon S3

Mountpoint para Amazon S3 es un cliente de archivos de código abierto de alto rendimiento para montar un bucket de Amazon S3 como un sistema de archivos local. Con Mountpoint, sus aplicaciones pueden acceder a objetos almacenados en Amazon S3 mediante operaciones del sistema de archivos, como abrir y leer. Mountpoint traduce automáticamente estas operaciones en llamadas a la API de objetos de S3, lo que proporciona a sus aplicaciones acceso al almacenamiento elástico y al rendimiento de Amazon S3 a través de una interfaz de archivos.

Mountpoint para Amazon S3 está disponible de forma general para su uso en producción en sus aplicaciones de lectura intensiva a gran escala: lagos de datos, formación de machine learning, renderización de imágenes, simulación de vehículos autónomos, extracción, transformación y carga (ETL), etc.

Mountpoint admite las operaciones básicas del sistema de archivos y puede leer archivos de hasta 5 TB. Puede enumerar y leer los archivos existentes y puede crear otros nuevos. No puede modificar los archivos existentes ni eliminar directorios, y no admite enlaces simbólicos ni el bloqueo de archivos. Mountpoint es ideal para aplicaciones que no necesitan todas las características de un sistema de archivos compartido y permisos tipo POSIX, pero que requieren el rendimiento elástico de Amazon S3 para leer y escribir grandes conjuntos de datos S3. Para obtener más detalles, consulte Mountpoint file system behavior (Comportamiento del sistema de archivos Mountpoint) en GitHub. En el caso de cargas de trabajo que requieran compatibilidad total con POSIX, recomendamos Amazon FSx para Lustre y su compatibilidad con la vinculación de buckets de S3.

Mountpoint para Amazon S3 solo está disponible para sistemas operativos Linux. Puede utilizar Mountpoint para acceder a objetos S3 en todas las clases de almacenamiento excepto S3 Glacier Flexible Retrieval, S3 Glacier Deep Archive, S3 Intelligent-Tiering Archive Access Tier y S3 Intelligent-Tiering Deep Archive Access Tier.

Temas

Instalación de Mountpoint

Puede descargar e instalar paquetes prediseñados de Mountpoint para Amazon S3 mediante la línea de comandos. Las instrucciones para descargar e instalar Mountpoint varían en función del sistema operativo Linux que utilice.

Temas

Distribuciones basadas en RPM (Amazon Linux, Fedora, CentOS, RHEL)

1. Copie la siguiente URL de descarga correspondiente a su arquitectura.

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. Descargue el paquete de Mountpoint para Amazon S3. Reemplace *download-link* por la URL de descarga apropiada del paso anterior.

```
wget download-link
```

3. (Opcional) Verifique la autenticidad e integridad del archivo descargado. En primer lugar, copie la URL de firma adecuada para su arquitectura.

   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
```

   A continuación, consulte Verificación de la firma del paquete de Mountpoint para Amazon S3.

4. Instale el paquete mediante el siguiente comando:

```
sudo yum install ./mount-s3.rpm
```

5. Verifique que Mountpoint se ha instalado correctamente; para ello, introduzca el siguiente comando:

```
mount-s3 --version
```

   Debería ver una salida similar a esta:

```
mount-s3 1.3.1
```

Distribuciones basadas en DEB (Debian,Ubuntu)

1. Copie la URL de descarga correspondiente a su arquitectura.

   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. Descargue el paquete de Mountpoint para Amazon S3. Reemplace *download-link* por la URL de descarga apropiada del paso anterior.

   ```
   wget download-link
   ```

3. (Opcional) Verifique la autenticidad e integridad del archivo descargado. En primer lugar, copie la URL de firma para su arquitectura.

   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
   ```

   A continuación, consulte Verificación de la firma del paquete de Mountpoint para Amazon S3.

4. Instale el paquete mediante el siguiente comando:

   ```
   sudo apt-get install ./mount-s3.deb
   ```

5. Verifique que Mountpoint para Amazon S3 se ha instalado correctamente mediante la ejecución del siguiente comando:

   ```
   mount-s3 --version
   ```

   Debería ver una salida similar a esta:

```
mount-s3 1.3.1
```

Otras distribuciones de Linux

1. Consulte la documentación de su sistema operativo para instalar los paquetes FUSE y `libfuse2` necesarios.
2. Copie la URL de descarga correspondiente a su arquitectura.

   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. Descargue el paquete de Mountpoint para Amazon S3. Reemplace *`download-link`* por la URL de descarga apropiada del paso anterior.

   ```
   wget download-link
   ```

4. (Opcional) Verifique la autenticidad e integridad del archivo descargado. En primer lugar, copie la URL de firma para su arquitectura.

   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
   ```

   A continuación, consulte Verificación de la firma del paquete de Mountpoint para Amazon S3.

5. Instale el paquete mediante el siguiente comando:

```
sudo mkdir -p /opt/aws/mountpoint-s3 && sudo tar -C /opt/aws/mountpoint-s3 -xzf ./
mount-s3.tar.gz
```

6. Agregue el binario `mount-s3` a su variable de entorno `PATH`. En su archivo `$HOME/.profile`, añada la siguiente línea:

   ```
   export PATH=$PATH:/opt/aws/mountpoint-s3/bin
   ```

   Guarde el archivo `.profile` y ejecute el siguiente comando:

   ```
   source $HOME/.profile
   ```

7. Verifique que Mountpoint para Amazon S3 se ha instalado correctamente mediante la ejecución del siguiente comando:

   ```
   mount-s3 --version
   ```

   Debería ver una salida similar a esta:

   ```
   mount-s3 1.3.1
   ```

Verificación de la firma del paquete Mountpoint para Amazon S3

1. Instale GnuPG (el comando `gpg`). Se requiere para verificar la autenticidad y la integridad de un paquete descargado de Mountpoint para Amazon S3. GnuPG se instala de forma predeterminada en las imágenes de máquina de Amazon (AMI) de Amazon Linux. Tras la instalación de GnuPG, vaya al paso 2.
2. Descargue la clave pública de Mountpoint mediante el siguiente comando:

   ```
   wget https://s3.amazonaws.com/mountpoint-s3-release/public_keys/KEYS
   ```

3. Importe la clave pública de Mountpoint a su llavero mediante el siguiente comando:

   ```
   gpg --import KEYS
   ```

4. Verifique la huella digital de la clave pública de Mountpoint mediante el siguiente comando:

```
gpg --fingerprint mountpoint-s3@amazon.com
```

Confirme que la cadena de huella mostrada coincide con la siguiente:

```
673F E406 1506 BB46 9A0E  F857 BE39 7A52 B086 DA5A
```

Si la cadena de huella no coincide, no termine de instalar Mountpoint y contacte con AWS Support.

5. Descargue el archivo de firma de paquetes. Reemplace *signature-link* por el enlace de firma apropiado de las secciones anteriores.

```
wget signature-link
```

6. Verifique la firma del paquete descargado mediante el siguiente comando. Reemplace *signature-filename* por el nombre de archivo del paso anterior.

```
gpg --verify signature-filename
```

Por ejemplo, en las distribuciones basadas en RPM, como Amazon Linux, introduzca el siguiente comando:

```
gpg --verify mount-s3.rpm.asc
```

7. El resultado debe incluir la frase `Good signature`. Si el resultado incluye la frase `BAD signature`, vuelva a descargar el archivo del paquete de Mountpoint y repita estos pasos. Si el problema persiste, no termine de instalar Mountpoint y contacte con AWS Support.

   El resultado puede incluir una advertencia sobre una firma de confianza. Esto no indica ningún problema. Solo significa que no ha verificado de forma independiente la clave pública de Mountpoint.

Configuración y uso de Mountpoint

Para utilizar Mountpoint para Amazon S3, su host necesita credenciales de AWS válidas con acceso al bucket o buckets que desea montar. Para conocer las distintas formas de autenticación, consulte Credenciales de AWS de Mountpoint en GitHub.

Por ejemplo, puede crear un usuario y rol de AWS Identity and Access Management (IAM) nuevos para este fin. Asegúrese de que este rol tiene acceso al bucket o buckets que desea montar. Puede pasar el rol de IAM a su instancia de Amazon EC2 con un perfil de instancia.

Uso de Mountpoint para Amazon S3

Utilice Mountpoint para Amazon S3 para hacer lo siguiente:

1. Montar buckets con el comando `mount-s3`.

   En el siguiente ejemplo, reemplace *DOC-EXAMPLE-BUCKET* por el nombre de su bucket de S3 y reemplace *~/mnt* por el directorio de su host en el que desea que se monte su bucket de S3.

   ```
   mkdir ~/mnt
   mount-s3 DOC-EXAMPLE-BUCKET ~/mnt
   ```

   Dado que el cliente de Mountpoint se ejecuta de forma predeterminada en segundo plano, el directorio *~/mnt* le concede ahora acceso a los objetos de su bucket de S3.
2. Acceda a los objetos de su bucket a través de Mountpoint.

   Después de montar su bucket localmente, puede utilizar comandos de Linux comunes, como `cat` o `ls`, para trabajar con sus objetos de S3. Mountpoint para Amazon S3 interpreta las claves de su bucket de S3 como rutas del sistema de archivos dividiéndolas en el carácter de barra diagonal (`/`). Por ejemplo, si tiene la clave de objeto `Data/2023-01-01.csv` en su bucket, tendrá un directorio llamado `Data` en su sistema de archivos de Mountpoint, con un archivo denominado `2023-01-01.csv` en él.

   Mountpoint para Amazon S3 no implementa intencionadamente la especificación completa del estándar POSIX para sistemas de archivos. Mountpoint se ha optimizado para cargas de trabajo que necesitan un acceso de lectura y escritura de alto rendimiento a los datos almacenados en Amazon S3 a través de una interfaz de sistema de archivos, pero que, por lo demás, no dependen de las características del sistema de archivos. Para obtener más información, consulte el comportamiento del sistema de archivos Mountpoint para Amazon S3 en GitHub. Los clientes que necesiten una semántica más completa del sistema de archivos deben considerar otros servicios de archivos de AWS, como Amazon Elastic File System (Amazon EFS) o Amazon FSx.
3. Desmonte su bucket mediante el comando `umount`. Este comando desmonta su bucket de S3 y sale de Mountpoint.

Para utilizar el siguiente comando de ejemplo, reemplace *~/mnt* por el directorio de su host en el que esté montado su bucket de S3.

```
umount ~/mnt
```

> **Note**
>
> Para obtener una lista de opciones para este comando, ejecute `umount --help`.

Para obtener más detalles sobre la configuración de Mountpoint, consulte Configuración del bucket de S3 y Configuración del sistema de archivos en GitHub.

Configuración del almacenamiento en caché en Mountpoint

Cuando utiliza Mountpoint para Amazon S3, puede configurarlo para almacenar en caché los datos a los que se ha accedido más recientemente desde sus buckets de S3 en el almacenamiento de instancias de Amazon EC2 o en un volumen de Amazon EBS adjunto. El almacenamiento en caché de estos datos puede ayudar a acelerar el rendimiento y reducir el coste del acceso repetido a los datos. El almacenamiento en caché en Mountpoint es ideal para casos de uso en los que se leen repetidamente los mismos datos que no cambian durante las múltiples lecturas. Por ejemplo, puede utilizar el almacenamiento en caché con trabajos de entrenamiento de machine learning que necesiten leer un conjunto de datos de entrenamiento varias veces para mejorar la precisión del modelo.

Cuando monta un bucket de S3, puede activar opcionalmente el almacenamiento en caché mediante marcas. Puede configurar la ubicación y el tamaño de la caché de datos y la cantidad de tiempo que los metadatos conservan en la memoria caché. Cuando monta un bucket y el almacenamiento en caché está activado, Mountpoint crea un subdirectorio vacío en la ubicación de caché configurada, si ese subdirectorio aún no existe. Al montar un bucket por primera vez y al desmontarlo, Mountpoint elimina el contenido de la ubicación de la caché. Para obtener más información sobre la configuración y el uso del almacenamiento en caché en Mountpoint, consulte Mountpoint for Amazon S3 Caching configuration en GitHub.

Cuando monta un bucket de S3, puede activar el almacenamiento en caché con la marca `--cache` *CACHE_PATH*. En el siguiente ejemplo, reemplace *CACHE_PATH* por la ruta de archivo al directorio en el que quiere almacenar en caché los datos. Reemplace *DOC-EXAMPLE-BUCKET* por el nombre

de su bucket de S3 y reemplace *~/mnt* por el directorio de su host en el que desea que se monte su bucket de S3.

```
mkdir ~/mnt
mount-s3 --cache CACHE_PATH DOC-EXAMPLE-BUCKET ~/mnt
```

> **Important**
>
> Si activa el almacenamiento en caché, Mountpoint conservará el contenido de los objetos no cifrados de su bucket de S3 en la ubicación de almacenamiento en caché configurada en el montaje. Para proteger sus datos, le recomendamos que restrinja el acceso a la ubicación de la caché de datos.

Solución de problemas de Mountpoint

Mountpoint para Amazon S3 cuenta está respaldado por AWS Support. Si necesita ayuda, póngase en contacto con el Centro de AWS Support.

También puede consultar y enviar problemas de Mountpoint en GitHub.

Si descubre un posible problema de seguridad en este proyecto, le rogamos que lo notifique a AWS Security a través de nuestra página de notificación de vulnerabilidades. No cree un problema público en GitHub.

Si su aplicación se comporta de forma inesperada con Mountpoint, puede inspeccionar la información de registro para diagnosticar el problema.

Registro

De forma predeterminada, Mountpoint emite información de registro de alta gravedad a syslog.

Para ver los registros en la mayoría de las distribuciones de Linux modernas, como Amazon Linux, ejecute el siguiente comando `journald`:

```
journalctl -e SYSLOG_IDENTIFIER=mount-s3
```

En otros sistemas Linux, es probable que las entradas `syslog` se escriban en un archivo como `/var/log/syslog`.

Puede utilizar estos registros para solucionar los problemas de su aplicación. Por ejemplo, si su aplicación intenta sobrescribir un archivo existente, se produce un error en la operación y verá una línea similar a la siguiente en el registro:

```
[WARN] open{req=12 ino=2}: mountpoint_s3::fuse: open failed: inode error: inode 2 (full
 key "README.md") is not writable
```

Para obtener más información, consulte Registro de Mountpoint para Amazon S3 en GitHub.

Configuración de transferencias de archivos rápidas y seguras con Amazon S3 Transfer Acceleration

Amazon S3 Transfer Acceleration es una característica de bucket que facilita la transferencia rápida, sencilla y segura de archivos a largas distancias entre su cliente y un bucket de S3. Transfer Acceleration está diseñada para optimizar las velocidades de transferencia en todo el mundo en buckets de S3. Transfer Acceleration aprovecha las ubicaciones de borde distribuidas globalmente en Amazon CloudFront. A medida que los datos llegan a una ubicación de borde, se redirigen a Amazon S3 a través de una ruta de red optimizada.

Por el uso de Transfer Acceleration se podrían aplicar cargos por transferencia de datos adicionales. Para obtener más información acerca de los precios, consulte Precios de Amazon S3.

¿Por qué utilizar Transfer Acceleration?

Puede que quiera usar Transfer Acceleration en un bucket por varios motivos:

- Porque sus clientes cargan a un bucket centralizado desde todo el mundo.
- Porque transfiere gigabytes o terabytes de datos regularmente entre varios continentes.
- Porque no puede utilizar todo el ancho de banda disponible en Internet al cargar a Amazon S3.

Para obtener más información acerca de cuándo usar Transfer Acceleration, consulte las preguntas frecuentes de Amazon S3.

Requisitos para utilizar Transfer Acceleration

Estos son los requisitos para usar Transfer Acceleration en un bucket de S3:

- Transfer Acceleration solo se admite en solicitudes de estilo alojadas virtualmente. Para obtener más información acerca de las solicitudes de estilo de alojamiento virtual, consulte Realizar solicitudes con la API REST.
- El nombre del bucket que use para Transfer Acceleration debe cumplir con las convenciones del DNS y no debe contener puntos (“.”).
- Transfer Acceleration debe estar activado en el bucket. Para obtener más información, consulte Habilitación y uso de S3 Transfer Acceleration.

  Tras habilitar Transfer Acceleration en un bucket, podría tardar hasta 20 minutos antes de que aumente la velocidad de transferencia de datos al bucket.

  > **Note**
  >
  > Aceleración de transferencia actualmente no es compatible con los buckets ubicados en las siguientes regiones:
  >
  > - Asia-Pacífico (Tokio) (ap-northeast-1)
  > - Asia-Pacífico (Seúl) (ap-northeast-2)
  > - Asia Pacífico (Bombay) (ap-south-1)
  > - Asia-Pacífico (Singapur) (ap-southeast-1)
  > - Asia-Pacífico (Sídney) (ap-southeast-2)
  > - Canadá (centro) (ca-central-1)
  > - Europa (Fráncfort) (eu-central-1)
  > - Europa (Irlanda) (eu-west-1)
  > - Europa (Londres) (eu-west-2)
  > - Europa (París) (eu-west-3)
  > - América del Sur (São Paulo) (sa-east-1)
  > - Este de EE. UU. (Norte de Virginia) (us-east-1)
  > - Este de EE. UU. (Ohio) (us-east-2)
  > - EE. UU. Oeste (Norte de California) (us-west-1)
  > - Oeste de EE. UU. (Oregón) (us-west-2)

- Para acceder al bucket habilitado para Transfer Acceleration, debe utilizar el punto de conexión *`bucketname`*`.s3-accelerate.amazonaws.com`. O bien, utilice el punto de conexión de doble pila *`bucketname`*`.s3-accelerate.dualstack.amazonaws.com` para conectar al bucket

habilitado por IPv6. Puede seguir utilizando los puntos de conexión habituales para la transferencia de datos estándar.

- Debe ser el propietario del bucket o establecer el estado de transfer acceleration. El propietario del bucket puede asignar permisos a otros usuarios para permitirles establecer el estado de aceleración en un bucket. El permiso `s3:PutAccelerateConfiguration` permite a los usuarios habilitar o deshabilitar Transfer Acceleration en un bucket. El permiso `s3:GetAccelerateConfiguration` permite a los usuarios devolver el estado de aceleración de transferencia de un bucket, que es `Enabled` o `Suspended`..

En las siguientes secciones se describe cómo utilizar Amazon S3 Transfer Acceleration para transferir datos.

Temas

- Introducción a Amazon S3 Transfer Acceleration
- Habilitación y uso de S3 Transfer Acceleration
- Uso de la herramienta Comparación de velocidad de Amazon S3 Transfer Acceleration

Introducción a Amazon S3 Transfer Acceleration

Puede utilizar Amazon S3 Transfer Acceleration para la transferencia rápida, sencilla y segura de archivos a largas distancias entre su cliente y un bucket de Amazon S3. Transfer Acceleration usa las ubicaciones de borde distribuidas globalmente en Amazon CloudFront. A medida que los datos llegan a una ubicación de borde, se redirigen a Amazon S3 a través de una ruta de red optimizada.

Para empezar a utilizar Amazon S3 Transfer Acceleration, siga estos pasos:

1. Activar Transfer Acceleration en un bucket

   Puede habilitar Transfer Acceleration en un bucket de cualquiera de las siguientes formas:

   - Uso de la consola de Amazon S3.
   - Uso de la operación PUT Bucket accelerate de la API de REST.
   - Use la AWS CLI y los SDK de AWS. Para obtener más información, consulte Desarrollo con Amazon S3 mediante los SDK de AWS.

   Para obtener más información, consulte Habilitación y uso de S3 Transfer Acceleration.

> **Note**
>
> Para que el bucket funcione con la aceleración de transferencia, el nombre del bucket debe ajustarse a los requisitos de nomenclatura de DNS y no debe contener puntos (“.”).

2. Transferir datos desde y hacia el bucket con aceleración habilitada

   Utilice uno de los siguientes nombres de dominio de punto de conexión s3-accelerate:

   - Para obtener acceso a un bucket con aceleración habilitada, use *`bucketname`*`.s3-accelerate.amazonaws.com`.
   - Para obtener acceso a un bucket con aceleración activada por IPv6, use *`bucketname`*`.s3-accelerate.dualstack.amazonaws.com`.

     Los puntos de enlace de doble pila de Amazon S3; admiten solicitudes a buckets de S3 a través de IPv6 y de IPv4. El punto de conexión de doble pila de Transfer Acceleration solo usa el tipo de nombre de punto de conexión de estilo de alojamiento virtual. Para obtener más información, consulte Introducción a la realización de solicitudes en IPv6 y Uso de puntos de conexión de doble pila en Amazon S3.

   > **Note**
   >
   > Su aplicación de transferencia de datos debe usar uno de los dos tipos de puntos de conexión siguientes para acceder al bucket y así acelerar la transferencia de datos: `.s3-accelerate.amazonaws.com` o `.s3-accelerate.dualstack.amazonaws.com` para el punto de conexión de doble pila. Si desea utilizar la transferencia de datos estándar, puede seguir utilizando los puntos de conexión habituales.

   Puede apuntar sus solicitudes PUT object y GET object en Amazon S3 al nombre de dominio de punto de conexión `s3-accelerate` después de habilitar Transfer Acceleration. Por ejemplo, supongamos que actualmente tiene una aplicación API de REST con PUT object que utiliza el nombre de host `mybucket.s3.us-east-1.amazonaws.com` en la solicitud PUT. Para acelerar PUT, cambie el nombre de host en su solicitud a `mybucket.s3-accelerate.amazonaws.com`. Para volver a utilizar la velocidad de carga estándar, vuelva a cambiar el nombre a `mybucket.s3.us-east-1.amazonaws.com`.

Después de habilitar Transfer Acceleration, puede tardar hasta 20 minutos en darse cuenta de los beneficios de desempeño. Sin embargo, el punto de conexión acelerado está disponible en cuando habilite Transfer Acceleration.

Puede utilizar el punto de conexión de aceleración en la AWS CLI, los SDK de AWS y otras herramientas que transfieren datos desde y hacia Amazon S3. Si utiliza los AWS SDK, algunos de los lenguajes admitidos usan una marca de configuración de cliente de punto de conexión acelerado, de modo que no tiene que establecer explícitamente el punto de conexión de Transfer Acceleration en *bucketname*`.s3-accelerate.amazonaws.com`. Para ver ejemplos de cómo usar un marcador de configuración de cliente de punto de conexión acelerado, consulte Habilitación y uso de S3 Transfer Acceleration.

Puede utilizar todas las operaciones de Amazon S3 a través de los puntos de enlace de aceleración de transferencia, excepto las siguientes:

- GET Service (list buckets)
- PUT Bucket (create bucket)
- DELETE Bucket

Además, Amazon S3 Transfer Acceleration no es compatible con las copias entre regiones usando PUT Object - Copy.

Habilitación y uso de S3 Transfer Acceleration

Puede utilizar Amazon S3 Transfer Acceleration para la transferencia rápida y segura de archivos a largas distancias entre su cliente y un bucket de S3. Puede habilitar Transfer Acceleration a través de la consola de S3, AWS Command Line Interface (AWS CLI) o los AWS SDK.

En esta sección se facilitan ejemplos de cómo habilitar Amazon S3 Transfer Acceleration en un bucket y usar el punto de conexión de aceleración para el bucket activado.

Para obtener más información acerca de los requisitos de Transfer Acceleration, consulte Configuración de transferencias de archivos rápidas y seguras con Amazon S3 Transfer Acceleration.

Uso de la consola de S3

> **Note**
>
> Si desea comparar velocidades de subida aceleradas y no aceleradas, abra la herramienta de comparación de velocidad de Amazon S3 Transfer Acceleration.
> La herramienta de comparación de velocidad utiliza cargas multiparte para transferir un archivo desde su navegador hacia diversas Regiones de AWS con Amazon S3 Transfer Acceleration y sin esta característica. Puede comparar la velocidad de subida para las subidas directas y transferir las subidas aceleradas por región.

Para habilitar Transfer Acceleration para un bucket de S3

1. Inicie sesión AWS Management Console Management Console y abra la consola de Amazon S3 en https://console.aws.amazon.com/s3/.
2. En la lista Buckets, seleccione el nombre del bucket para el que desea habilitar la aceleración de transferencia.
3. Seleccione Propiedades.
4. En Transfer acceleration (Aceleración de transferencia), elija Edit (Editar).
5. Elija Enable (Habilitar) y Save Changes (Guardar cambios).

Para acceder a transferencias aceleradas de datos

1. Una vez que Amazon S3 habilite la aceleración de transferencia para el bucket, consulte la pestaña Properties (Propiedades) del bucket.
2. En Transfer acceleration (Aceleración de transferencia), Accelerated endpoint (Punto de conexión acelerado) muestra el punto de conexión de aceleración de transferencia del bucket. Utilice este punto de conexión para acceder a transferencias de datos aceleradas desde y hacia el bucket.

   Si suspende Transfer Acceleration, el punto de conexión de aceleración deja de funcionar.

Uso de la AWS CLI

A continuación, se presentan ejemplos de comandos de la AWS CLI utilizados para Transfer Acceleration. Para obtener instrucciones acerca de cómo configurar la AWS CLI, consulte Desarrollo con Amazon S3 mediante la AWS CLI.

Habilitación de Transfer Acceleration en un bucket

Puede utilizar el comando put-bucket-accelerate-configuration de la AWS CLI para habilitar o suspender Transfer Acceleration en un bucket.

En el siguiente ejemplo se usa `Status=Enabled` para habilitar Transfer Acceleration en un bucket. Se utiliza `Status=Suspended` para suspender Transfer Acceleration.

Example

```
$ aws s3api put-bucket-accelerate-configuration --bucket bucketname --accelerate-configuration Status=Enabled
```

Uso de Transfer Acceleration

Puede dirigir todas las solicitudes de Amazon S3 realizadas por los comandos de la AWS CLI s3 y s3api al punto de conexión de aceleración: `s3-accelerate.amazonaws.com`. Para ello, establezca el valor de configuración `use_accelerate_endpoint` como `true` en un perfil en el archivo de AWS Config. Transfer Acceleration debe estar activado en su bucket para usar el punto de conexión acelerado.

Todas las solicitudes se envían mediante el direccionamiento al bucket de estilo virtual: `my-bucket.s3-accelerate.amazonaws.com`. Las solicitudes `ListBuckets`, `CreateBucket` y `DeleteBucket` no se envían al punto de conexión de aceleración, ya que este no admite estas operaciones.

Para obtener más información acerca de `use_accelerate_endpoint`, consulte Configuración de S3 con la AWS CLI en la Referencia de comandos de la AWS CLI.

En el siguiente ejemplo se establece `use_accelerate_endpoint` como `true` en el perfil predeterminado.

Example

```
$ aws configure set default.s3.use_accelerate_endpoint true
```

Si quiere usar el punto de conexión acelerado para algunos comandos de la AWS CLI pero no otros, puede usar uno de los dos siguientes métodos:

- Use el punto de conexión acelerado estableciendo el parámetro `--endpoint-url` como `https://s3-accelerate.amazonaws.com` para cualquier comando s3 o s3api.
- Configure perfiles separados en su archivo de AWS Config. Por ejemplo, puede crear un perfil que configure `use_accelerate_endpoint` como `true` y un perfil que no configure `use_accelerate_endpoint`. Al ejecutar un comando, especifique qué perfil quiere usar en función de si quiere usar el punto de conexión acelerado o no.

Cargar un objeto en un bucket habilitado para Transfer Acceleration

En el siguiente ejemplo se carga un archivo en un bucket activado para Transfer Acceleration usando el perfil predeterminado que se ha configurado para usar el punto de conexión acelerado.

Example

```
$ aws s3 cp file.txt s3://bucketname/keyname --region region
```

En el siguiente ejemplo se carga un archivo en un bucket activado para Transfer Acceleration usando el parámetro `--endpoint-url` para especificar el punto de conexión acelerado.

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
```

Uso de los AWS SDK

A continuación se presentan ejemplos del uso de Transfer Acceleration para cargar objetos en Amazon S3 con el AWS SDK. Algunos de los lenguajes compatibles con los AWS SDK (por ejemplo, Java y .NET) utilizan una marca de configuración de cliente de punto de conexión de aceleración, de modo que no tendrá que establecer de forma explícita el punto de conexión de Transfer Acceleration para *bucketname*.s3-accelerate.amazonaws.com.

Java

Example

En el siguiente ejemplo se demuestra cómo usar un punto de conexión acelerado para cargar un objeto en Amazon S3. En el ejemplo se realiza lo siguiente:

- Crea un `AmazonS3Client` que se configura para usar puntos de conexión de aceleración. Todos los buckets a los que accede el cliente deben tener Transfer Acceleration habilitado.
- Habilita Transfer Acceleration en un bucket específico. Este paso solo es necesario si el bucket que especifica aún no tiene Transfer Acceleration habilitado.
- Verifica que la aceleración de transferencia está habilitada para el bucket especificado.
- Carga un nuevo objeto en el bucket especificado utilizando el punto de conexión de aceleración del bucket.

Para obtener más información acerca del uso de Transfer Acceleration, consulte Introducción a Amazon S3 Transfer Acceleration. Para obtener instrucciones sobre cómo crear y probar una muestra funcional, consulte Introducción en la Guía del desarrollador de AWS SDK for Java.

```
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

En el siguiente ejemplo, se muestra cómo usar AWS SDK for .NET con el fin de habilitar Transfer Acceleration en un bucket. Para obtener información acerca de cómo configurar y ejecutar ejemplos de código, consulte Introducción al SDK de AWS para .NET en la Guía para desarrolladores del SDK de AWS para .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);
                }
        }
    }
}
```

Al cargar un objeto en un bucket que tenga Transfer Acceleration habilitado, especifique el uso del punto de conexión de aceleración en el momento de crear un cliente.

```
var client = new AmazonS3Client(new AmazonS3Config
            {
                RegionEndpoint = TestRegionEndpoint,
                UseAccelerateEndpoint = true
            }
```

Javascript

Para ver un ejemplo de cómo habilitar Transfer Acceleration con el AWS SDK para JavaScript, consulte Llamada a la operación putBucketAccelerateConfiguration en la Referencia de la API de AWS SDK para JavaScript.

Python (Boto)

Para obtener un ejemplo de cómo habilitar Transfer Acceleration con el SDK for Python, consulte put_bucket_accelerate_configuration en la Referencia de la API de AWS SDK for Python (Boto3).

Other

Para obtener información acerca del uso de otros AWS SDK, consulte Código de muestra y bibliotecas.

Uso de la API de REST

Utilice la operación `PutBucketAccelerateConfiguration` de la API de REST para habilitar la configuración acelerada en un bucket existente.

Para obtener más información, consulte PutBucketAccelerateConfiguration en la Referencia de la API de Amazon Simple Storage Service.

Uso de la herramienta Comparación de velocidad de Amazon S3 Transfer Acceleration

Puede utilizar la herramienta de comparación de velocidad de Amazon S3 Transfer Acceleration para comparar velocidades de subida aceleradas y no aceleradas en las regiones de Amazon S3. La herramienta Comparación de velocidad utiliza cargas multipartes para transferir un archivo desde su navegador hacia diversas regiones de Amazon S3 con y sin Transfer Acceleration.

Puede obtener acceso a la herramienta Comparación de velocidad utilizando cualquiera de los siguientes métodos:

- Copie la siguiente URL en la ventana de su navegador y sustituya *`region`* por la Región de AWS que utiliza (por ejemplo, `us-west-2`) y *`yourBucketName`* por el nombre del bucket que quiere evaluar:

  ```
  https://s3-accelerate-speedtest.s3-accelerate.amazonaws.com/en/
  accelerate-speed-comparsion.html?
  region=region&origBucketName=yourBucketName
  ```

  Para obtener una lista de las regiones compatibles con Amazon S3, consulte Puntos de conexión y cuotas de Amazon S3 en la Referencia general de AWS.
- Uso de la consola de Amazon S3.

Utilización de buckets de pago por solicitante para transferencias de almacenamiento y uso

En general, los propietarios de buckets pagan todos los costos de almacenamiento y transferencia de datos de Amazon S3 asociados con el bucket. Sin embargo, puede configurar un bucket para que sea un bucket de pago por solicitante . Con los buckets de pago por solicitante, el solicitante, en

lugar del propietario del bucket, paga el costo de la solicitud y de la descarga de datos del bucket. El propietario del bucket siempre paga el costo de almacenamiento de datos.

Por lo general, los buckets se configuran para que sean buckets de pago por solicitante cuando desea compartir datos pero no incurrir en cargos asociados con otros que acceden a los datos. Por ejemplo, puede utilizar los buckets de pago por solicitante al poner a disposición conjuntos de datos grandes, como directorios de código postal, datos de referencia, información geoespacial o datos de rastreo web.

> **Important**
>
> Si habilita los pagos por solicitante en un bucket, no se permite el acceso anónimo a ese bucket.

Debe autenticar todas las solicitudes relacionadas con buckets de pago por solicitante. La autenticación de la solicitud le permite a Amazon S3 identificar y cobrarle al solicitante el uso del bucket de pago por solicitante.

Cuando el solicitante asume un rol de AWS Identity and Access Management (IAM) antes de realizar la solicitud, la solicitud se le cobra a la cuenta a la que pertenece el rol. Para obtener más información acerca de los roles de IAM, consulte Funciones de IAM en la Guía del usuario de IAM.

Después de configurar un bucket para que sea un bucket de pago por solicitante, los solicitantes deben mostrar que comprenden que se les cobrará por la solicitud y la descarga de datos. Para demostrar que aceptan los cargos, los solicitantes deben incluir `x-amz-request-payer` en el encabezado de su solicitud de la API para las solicitudes DELETE, GET, HEAD, POST y PUT o agregar el parámetro `RequestPayer` en su solicitud REST. Para las solicitudes a la CLI, los solicitantes pueden usar el parámetro `--request-payer`.

Example — Usar el pago por solicitante al eliminar un objeto

Para utilizar el siguiente ejemplo de API DeleteObjectVersion, sustituya *`user input placeholders`* por su información.

```
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 el solicitante restaura los objetos mediante la API RestoreObject, se admite el pago por solicitante si el encabezado `x-amz-request-payer` o el parámetro `RequestPayer` están en la solicitud; sin embargo, el solicitante solo paga el costo de la solicitud. El propietario del bucket es el que paga los gastos de recuperación.

Los buckets de pago por solicitante no admiten lo siguiente:

- Solicitudes anónimas
- Solicitudes SOAP
- Uso de un bucket de pago por solicitante como bucket de destino para el registro del usuario final, o viceversa. Sin embargo, puede activar el registro de usuario final en un bucket de pago por solicitante en el que el bucket de destino no sea un bucket de pago por solicitante.

Cómo funcionan los pagos por solicitante

El cargo por solicitudes de pago por solicitante correcto es sencillo: el solicitante paga por la transferencia de datos y la solicitud y el propietario del bucket paga por el almacenamiento de datos. Sin embargo, el propietario del bucket paga la solicitud en las siguientes condiciones:

- La solicitud devuelve un error `AccessDenied` (HTTP `403 Forbidden`) y se inicia dentro de la cuenta de AWS individual del propietario del bucket o de la organización de AWS.
- La solicitud es una solicitud de Simple Object Access Protocol (SOAP, Protocolo simple de acceso a objetos).

Para obtener más información acerca del pago por solicitante, consulte los siguientes temas.

Temas

- Configuración de pago por solicitante en un bucket
- Recuperación de la configuración RequestPayment mediante la API de REST
- Descarga de objetos desde buckets de pago por solicitante

Configuración de pago por solicitante en un bucket

Puede configurar un bucket de Amazon S3 para que sea un bucket de pago por solicitante, de este modo el solicitante paga el costo de la solicitud y la descarga de datos en lugar del propietario del bucket.

Esta sección proporciona ejemplos de cómo configurar el pago por solicitante en un bucket de Amazon S3 mediante la consola y la API REST.

Uso de la consola de S3

Para habilitar el pago por solicitante para un bucket de S3

1. Inicie sesión en la AWS Management Console y abra la consola de Amazon S3 en https://console.aws.amazon.com/s3/.
2. En la lista Buckets (Buckets), elija el nombre del bucket para el que desea habilitar el pago por solicitante.
3. Seleccione Properties (Propiedades).
4. En Requester pays (Pago por solicitante), seleccione Edit (Editar).
5. Elija Enable (Habilitar) y Save Changes (Guardar cambios).

   Amazon S3 habilita el pago por solicitante para su bucket y muestra la Bucket overview (Información general del bucket). En Pago por solicitante, verá Habilitado.

Uso de la API de REST

Sólo el propietario del bucket puede establecer el valor de `RequestPaymentConfiguration.payer` configuración de un bucket en `BucketOwner` (el valor predeterminado) o `Requester`. La configuración del recurso `requestPayment` es opcional. De forma predeterminada, el bucket no es un bucket de pago por solicitante.

Para que un bucket de pago por solicitante vuelva a ser un bucket normal, se usa el valor `BucketOwner`. Por lo general, se usaría el valor `BucketOwner` al cargar datos al bucket de Amazon S3 y luego se establecería el valor en `Requester` antes de publicar objetos en el bucket.

Para configurar el recurso requestPayment

- Use una solicitud PUT para establecer el valor `Payer` en `Requester` en un bucket especificado.

```
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 solicitud se realiza correctamente, Amazon S3 devuelve una respuesta similar a la siguiente:

```
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
```

Puede establecer el pago por solicitante sólo en el nivel del bucket. No se puede establecer el pago por solicitante para objetos específicos dentro del bucket.

Puede configurar un bucket para que tenga el valor `BucketOwner` o `Requester` en cualquier momento. Sin embargo, puede haber unos minutos antes de que el nuevo valor de configuración surta efecto.

> **Note**
>
> Los propietarios de bucket que dan URL prefirmadas deben considerarlo cuidadosamente antes de configurar un bucket para que sea pago por solicitante, especialmente si la URL tiene una larga vida útil. Al propietario del bucket se le cobra cada vez que el solicitante usa una URL prefirmada que usa las credenciales del propietario del bucket.

Recuperación de la configuración RequestPayment mediante la API de REST

Puede determinar el valor `Payer` que se establece en un bucket mediante la solicitud del recurso `requestPayment`.

Para obtener el recurso requestPayment

- Use una solicitud GET para obtener el recurso `requestPayment`, como se muestra en la siguiente solicitud.

```
GET ?requestPayment HTTP/1.1
Host: [BucketName].s3.amazonaws.com
Date: Wed, 01 Mar 2009 12:00:00 GMT
Authorization: AWS [Signature]
```

Si la solicitud se realiza correctamente, Amazon S3 devuelve una respuesta similar a la siguiente:

```
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>
```

Esta respuesta muestra que el valor `payer` está configurado en `Requester`.

Descarga de objetos desde buckets de pago por solicitante

Debido a que se les cobra la descarga de datos de los buckets de pago por solicitante, las solicitudes deben contener un parámetro especial `x-amz-request-payer`, que confirma que el solicitante sabe que se les cobrará por la descarga. Para obtener acceso a los objetos en los buckets de pago por solicitante, las solicitudes deben incluir lo siguiente:

- Para las solicitudes DELETE, GET, HEAD, POST y PUT, incluya `x-amz-request-payer : requester` en el encabezado
- Para los URL firmados, incluya `x-amz-request-payer=requester` en la solicitud.

Si la solicitud se realiza correctamente y se le cobra al solicitante, la respuesta incluye el encabezado `x-amz-request-charged:requester`. Si `x-amz-request-payer` no se encuentra en la solicitud, Amazon S3 devuelve el error 403 y le cobra la solicitud al propietario del bucket.

> **Note**
>
> Los propietarios de buckets no necesitan añadir `x-amz-request-payer` a sus solicitudes. Asegúrese de que haya incluido `x-amz-request-payer` y su valor en el cálculo de firmas. Para obtener más información, consulte Construcción del elemento CanonicalizedAmzHeaders.

Uso de la API de REST

Para descargar objetos de un bucket de pago por solicitante

- Use una solicitud `GET` para descargar un objeto de un bucket de pago por solicitante, como se muestra en la siguiente solicitud.

```
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]
```

Si la solicitud GET se realiza correctamente y se le cobra al solicitante, la respuesta incluye `x-amz-request-charged:requester`.

Amazon S3 puede devolver el error `Access Denied` en el caso de solicitudes que intentan obtener objetos de un bucket de pago por solicitante. Para obtener más información, consulte Respuestas de error en la Referencia de API de Amazon Simple Storage Service.

Mediante AWS CLI

Para descargar objetos de un bucket de pago por solicitante mediante la AWS CLI, especifique `--request-payer requester` como parte de la solicitud `get-object`. Para obtener más información, consulte get-object en la Referencia de la AWS CLI.

Cuotas, restricciones y limitaciones de bucket

Cada bucket de Amazon S3 es propiedad de la Cuenta de AWS que lo creó. La propiedad del bucket no se puede transferir a otra cuenta.

Límites de cuotas de bucket

De forma predeterminada, puede crear hasta 100 buckets en cada una de sus Cuentas de AWS. Si necesita buckets adicionales, puede aumentar la cuota de buckets de su cuenta hasta un máximo de 1000 buckets mediante el envío de una solicitud de aumento de cuota. No hay diferencia en el rendimiento cuando usa muchos buckets o solo unos pocos.

> **Note**
>
> No es necesario enviar varias solicitudes de aumento de cuota para cada Región de AWS. La cuota de buckets se aplica a su Cuenta de AWS.

Para obtener información sobre cómo aumentar su cuota de bucket, consulte Amazon S3 endpoints and quotas en la Referencia general de Amazon Web Services.

Limitaciones de buckets y objetos

No hay un límite o tamaño máximo para el número de objetos que se puede almacenar en un bucket. Puede almacenar todos los objetos en un solo bucket u organizarlos en varios buckets. Sin embargo, no se puede crear un bucket desde dentro de otro bucket.

Límites de nombres de buckets

Cuando cree un bucket, elija su nombre y la Región de AWS en la que se creará. Una vez que haya creado un bucket, no podrá modificar su nombre ni su región.

Al nombrar un bucket, elija un nombre que sea relevante para usted o su empresa. Evite el uso de nombres asociados con otros. Por ejemplo, debe evitar usar `AWS` o `Amazon` en el nombre del bucket.

Reutilización de los nombres de bucket

Si un bucket está vacío, puede eliminarlo. Después de eliminar un bucket, el nombre vuelve a estar disponible para su reutilización. Sin embargo, después de eliminar el bucket, es posible que no pueda volver a utilizar el nombre por varios motivos.

Por ejemplo, cuando elimina el bucket y el nombre queda disponible para su reutilización, otra Cuenta de AWS podría crear un bucket con ese nombre. Además, puede pasar algún tiempo antes de que pueda volver a utilizar el nombre de un bucket eliminado. Si desea utilizar el mismo nombre de bucket, le recomendamos que no elimine el bucket.

Para obtener más información acerca de los nombres de bucket, consulte Reglas de nomenclatura de buckets.

Nomenclatura de buckets y buckets creados automáticamente

Si su aplicación crea buckets automáticamente, elija un esquema de nomenclatura de buckets que sea poco probable que ocasione conflictos de nomenclatura. Asegúrese de que la lógica de su aplicación elija un nombre de bucket diferente si un nombre de bucket ya ha sido usado.

Para obtener más información sobre la nomenclatura de buckets, consulte Reglas de nomenclatura de buckets.

Operaciones con buckets

La ingeniería de alta disponibilidad de Amazon S3 se centra en las operaciones get, put, list y delete. Debido a que las operaciones de los buckets funcionan en un espacio de recursos centralizados y globales, no es apropiado crear, eliminar ni configurar buckets en la ruta del código de alta disponibilidad de su aplicación. Es mejor crear, eliminar o configurar buckets en una rutina de inicialización o configuración distinta que ejecute con menos frecuencia.

Cargar, descargar y trabajar con objetos en Amazon S3

Para almacenar datos en Amazon S3, trabaja con recursos conocidos como buckets y objetos. Un bucket es un contenedor de objetos. Un objeto es un archivo y cualquier metadato que describa ese archivo.

Para almacenar un objeto en Amazon S3, cree un bucket y, a continuación, cargue el objeto en el bucket. Cuando el objeto está en el bucket, puede abrirlo, descargarlo y copiarlo. Cuando ya no necesite un objeto o un bucket, puede limpiar estos recursos.

> **Note**
>
> Para obtener más información sobre el uso de la clase de almacenamiento Amazon S3 Express One Zone con buckets de directorio, consulte ¿Qué es S3 Express One Zone? y Buckets de directorio.

> **Important**
>
> En la consola de Amazon S3, al elegir Open (Abrir) o Download As (Descargar como) para un objeto, estas operaciones crean URL prefirmadas. Durante cinco minutos, cualquier persona que tenga acceso a estas URL prefirmadas podrá acceder a su objeto. Para obtener más información sobre las URL prefirmadas, consulte Uso de direcciones URL prefirmadas.

Con Amazon S3 paga únicamente por lo que usa. Para obtener más información acerca de las características y precios de Amazon S3, consulte Amazon S3. Si es cliente nuevo de Amazon S3, puede comenzar con Amazon S3 de forma gratuita. Para obtener más información, consulte Capa gratuita de AWS.

Temas

- Copia, traslado y cambio de nombre de objetos
- Descarga de objetos
- Comprobación de la integridad de objetos
- Eliminación de objetos de Amazon S3
- Organizar, describir y trabajar con los objetos
- Uso de URL prefirmadas
- Transformación de objetos con Lambda para objetos S3

Información general de los objetos de Amazon S3

Amazon S3 es un almacén de objetos que utiliza valores clave únicos para almacenar tantos objetos como desee. Almacene estos objetos en uno o más buckets. Cada objeto puede tener un tamaño de hasta 5 TB. Un objeto consiste en lo siguiente:

Key

El nombre que se asigna a un objeto. La clave de objeto se usa para recuperar el objeto. Para obtener más información, consulte Trabajar con metadatos de objeto.

ID de versión.

En un bucket, una clave y un ID de versión identifican exclusivamente un objeto. El ID de versión es una cadena que genera Amazon S3 cuando se agrega un objeto a un bucket. Para obtener más información, consulte Usar el control de versiones en buckets de S3.

Valor

El contenido que está almacenando.

El valor de un objeto puede ser cualquier secuencia de bytes. El tamaño de los objetos puede variar desde cero hasta 5 TB. Para obtener más información, consulte Carga de objetos.

Metadatos

Conjunto de pares nombre-valor con los que puede almacenar información relativa al objeto. Puede asignar metadatos, que se denominan metadatos definidos por el usuario, a sus objetos en Amazon S3. Amazon S3 asigna también metadatos de sistema a estos objetos, que se usan para administrar objetos. Para obtener más información, consulte Trabajar con metadatos de objeto.

Subrecursos

Amazon S3 utiliza el mecanismo de subrecursos para almacenar información adicional específica del objeto. Dado que los subrecursos están subordinados a los objetos, siempre están asociados con otras entidades, como objetos o buckets. Para obtener más información, consulte Subrecursos de objeto.

Información de control de acceso

Puede controlar el acceso a los objetos que almacena en Amazon S3. Amazon S3 admite tanto el control de acceso basado en recursos, como con una lista de control de acceso (ACL) o políticas de bucket, como el control de acceso basado en usuarios. Para obtener más información sobre el control de acceso, consulte lo siguiente:

- Administración de accesos
- Administración de identidades y accesos para Amazon S3
- Configuración de la ACL

Los recursos de Amazon S3 (por ejemplo: buckets y objetos) son privados de manera predeterminada. Es necesario conceder permisos explícitos para que otras personas puedan obtener acceso a estos recursos. Para obtener más información acerca de compartir objetos, consulte Uso compartido de objetos con URL prefirmadas.

Etiquetas

Puede usar etiquetas para clasificar los objetos almacenados, para el control de acceso o para la asignación de costos. Para obtener más información, consulte Categorización del almacenamiento mediante etiquetas.

Subrecursos de objeto

Amazon S3 define un conjunto de subrecursos asociados con buckets y objetos. Los subrecursos están subordinados a los objetos. Esto significa que los subrecursos no existen por sí solos. Siempre están asociados a alguna otra entidad, como un objeto o un bucket.

En la siguiente tabla se describen los subrecursos asociados objetos de Amazon S3.

Subrecurso	Descripción
acl	Contiene una lista de concesiones en la que se identifican los beneficiarios y los permisos concedidos. Al crear un objeto, la `acl` identifica al propietario del objeto que tiene control total sobre el mismo. Puede recuperar la ACL de un objeto o sustituirla por una lista actualizada de concesiones. Cualquier actualización en una ACL exige que sustituya la ACL existente. Para obtener más información acerca de las ACL, consulte Información general de las Listas de control de acceso (ACL).

Creación de nombres de clave de objeto

La clave de objeto (o nombre de clave) identifica de forma única el objeto en un bucket de Amazon S3. Los metadatos de objetos son un conjunto de pares nombre-valor. Para obtener más información acerca de los metadatos del objeto, consulte Trabajar con metadatos de objeto.

Al crear un objeto, especifica su nombre de clave, que identifica exclusivamente el objeto en el bucket. Por ejemplo, en la consola de Amazon S3, al destacar un bucket aparece una lista de objetos en el bucket. Esos nombres son las claves de objeto. El nombre de clave de objeto es una secuencia de caracteres Unicode con codificación UTF-8 de una longitud máxima de 1024 bytes. Los nombres de clave de objeto distinguen entre mayúsculas y minúsculas.

> **Note**
>
> Los nombres de clave de objeto con el valor "soap" no son compatibles con las solicitudes de tipo host virtual. Para los valores de los nombres de las claves de los objetos en los que se utiliza "soap", se debe utilizar en su lugar una URL de tipo ruta.

El modelo de datos de Amazon S3 es una estructura plana: usted crea un bucket y el bucket almacena objetos. No existe una jerarquía entre los subbuckets o las subcarpetas. Sin embargo, puede inferir una jerarquía lógica con prefijos de nombres de clave y delimitadores del mismo modo que lo hace la consola de Amazon S3. La consola de Amazon S3 admite el concepto de carpetas. Para obtener más información sobre cómo editar metadatos desde la consola de Amazon S3, consulte Edición de metadatos de objeto en la consola de Amazon S3.

Supongamos que el bucket (`admin-created`) tiene cuatro objetos con las siguientes claves de objeto:

`Development/Projects.xls`

`Finance/statement1.pdf`

`Private/taxdocument.pdf`

`s3-dg.pdf`

La consola utiliza los prefijos de nombre de clave (`Development/`, `Finance/` y `Private/`) y el delimitador ("/") para presentar una estructura de carpetas. La clave de `s3-dg.pdf` no tiene un prefix, por lo que su objeto aparece directamente en el nivel raíz del bucket. Si abre la carpeta `Development/`, verá el objeto `Projects.xlsx` en ella.

- Amazon S3 admite buckets y objetos y no hay jerarquía. Sin embargo, si utiliza prefijos y delimitadores en un nombre de clave de objeto, la consola de Amazon S3 y los SDK de AWS pueden inferir la jerarquía e introducir el concepto de carpetas.
- La consola de Amazon S3 implementa la creación de objetos de carpeta mediante la creación de objetos de cero bytes con el valor de prefijo y delimitador de carpeta como clave. Estos objetos de carpeta no aparecen en la consola. De lo contrario, se comportan como cualquier otro objeto y se pueden ver y manipular a través de la API de REST, la AWS CLI y los SDK de AWS.

Directrices de nomenclatura de claves de objeto

Puede usar cualquier carácter UTF-8 en un nombre de clave de objeto. Sin embargo, el uso de ciertos caracteres en los nombres de las claves puede provocar problemas con algunas aplicaciones y protocolos. Las siguientes directrices le ayudan a aumentar al máximo el cumplimiento con DNS, caracteres seguros para la web, analizadores XML y otras API.

Caracteres seguros

Los siguientes conjuntos de caracteres son habitualmente seguros para su uso en nombres de claves.

Alphanumeric characters	• 0-9 • a-z

	• A-Z
Special characters	• Signo de exclamación (!) • Guion (-) • Guion bajo (_) • Punto (.) • Asterisco (*) • Comilla simple (') • Abrir paréntesis (() • Cerrar paréntesis ())

A continuación se proporcionan ejemplos de nombres de claves de objeto válidos:

- `4my-organization`
- `my.great_photos-2014/jan/myvacation.jpg`
- `videos/2014/birthday/video1.wmv`

> **Note**
>
> En el caso de los objetos con nombres de clave que terminen con punto(s) “.” descargados mediante la consola de Amazon S3, se eliminarán los puntos “.” del nombre de clave del objeto descargado. Para descargar un objeto cuyo nombre de clave termine con punto(s) “.”, debe utilizar la AWS Command Line Interface (AWS CLI), los SDK de AWS o la API de REST.
>
> Además, tenga en cuenta las siguientes limitaciones de prefijo:
>
> - Los objetos con un prefijo de "./" se deben cargar o descargar con el AWS Command Line Interface (AWS CLI), AWS SDK o la API de REST. No puede usar la consola de Amazon S3.
> - Los objetos con un prefijo de "../" no se pueden cargar usando la herramienta AWS Command Line Interface (AWS CLI) o la consola de Amazon S3.

Caracteres que podrían requerir un trato especial

Los siguientes caracteres de un nombre de clave podrían requerir un trato adicional en cuando a codificación, y probablemente tengan que codificarse en la URL o haya que referirse a ellos en HEX. Algunos de ellos son caracteres no imprimibles que su navegador podría no admitir, por lo que también requieren un trato especial:

- Ampersand ("&")
- Dólar ("$")
- Rangos de caracteres ASCII 00–1F hex (0–31 decimal) y 7F (127 decimal)
- Arroba ("@")
- Igual ("=")
- Punto y coma (";")
- Barra inclinada ("/")
- Dos puntos (":")
- Más ("+")
- Espacio: puede que se pierdan secuencias significativas de espacios en algunos usos (especialmente espacios múltiples)
- Coma (",")
- Signo de cierre de interrogación ("?")

Caracteres que deben evitarse

Recomendamos que no utilice los siguientes caracteres en un nombre de clave debido a un manejo de caracteres significativamente especial, que no es coherente en todas las aplicaciones.

- Barra diagonal invertida ("\")
- Llave de apertura ("{")
- Caracteres ASCII no imprimibles (caracteres decimales 128-255)
- Acento circunflejo ("^")
- Llave de cierre ("}")
- Carácter de porcentaje ("%")
- Acento grave ("`")
- Corchete de cierre ("]")

- Comillas
- Símbolo mayor que (">")
- Corchete de apertura ("[")
- Tilde ("~")
- Símbolo menor que ("<")
- Almohadilla ("#")
- Barra vertical ("|")

Restricciones de clave de objeto relacionadas con XML

Según lo especificado por el estándar XML sobre el tratamiento de final de línea, todo el texto XML se normaliza de manera que los retornos de carro único (código ASCII 13) y los retornos de carro seguidos inmediatamente de un salto de línea (código ASCII 10) se sustituyen por un único carácter de salto de línea. Para garantizar el análisis correcto de las claves de objeto en las solicitudes XML, los retornos de carro y otros caracteres especiales deben reemplazarse por su código de entidad XML equivalente cuando se insertan dentro de etiquetas XML. A continuación se muestra una lista de estos caracteres especiales y sus códigos de entidad equivalentes:

- ' como `'`
- ” como `"`
- & como `&`
- < como `<`
- > como `>`
- \r como `
` o `
`
- \n como `
` o `
`

Example

En el ejemplo siguiente se ilustra el uso de un código de entidad XML como sustitución de un retorno de carro. Esta solicitud `DeleteObjects` elimina un objeto con el parámetro `key`: `/some/prefix/objectwith\rcarriagereturn` (donde \r es el retorno de carro).

```
<Delete xmlns="http://s3.amazonaws.com/doc/2006-03-01/">
  <Object>
    <Key>/some/prefix/objectwith&#13;carriagereturn</Key>
```

```
  </Object>
</Delete>
```

Trabajar con metadatos de objeto

Puede establecer metadatos de objeto en Amazon S3 en el momento de cargar el objeto. Los metadatos de objetos son un conjunto de pares nombre-valor. Tras cargar el objeto, no puede modificar sus metadatos. La única manera de modificar los metadatos de un objeto es realizar una copia del mismo y configurar sus metadatos.

Al crear un objeto, también se especifica el nombre de la clave, que identifica de forma única al objeto en el bucket. La clave de objeto (o nombre de clave) identifica de forma única el objeto en un bucket de Amazon S3. Para obtener más información, consulte Creación de nombres de clave de objeto.

Existen dos tipos de metadatos en Amazon S3: metadatos definidos por el sistema y metadatos definidos por el usuario. En las secciones siguientes se proporciona más información acerca de los metadatos definidos por el sistema y definidos por el usuario. Para obtener más información sobre cómo editar metadatos mediante la consola de Amazon S3, consulte Edición de metadatos de objeto en la consola de Amazon S3.

Metadatos de objetos definidos por el sistema

Para cada objeto almacenado en un bucket, Amazon S3 mantiene un conjunto de metadatos del sistema. Amazon S3 procesa estos metadatos del sistema según sea necesario. Por ejemplo, Amazon S3 mantiene la fecha de creación del objeto y los metadatos de tamaño y usa esta información como parte de la administración del objeto.

Existen dos categorías de metadatos del sistema:

- Controlados por el sistema: los metadatos, como la fecha de creación del objeto, están controlados por el sistema, y solo Amazon S3 puede modificar su valor.
- Controlados por el usuario: otros metadatos de sistema, como la clase de almacenamiento configurada para el objeto y si el objeto tiene habilitado el cifrado del lado del servidor, son ejemplos de metadatos del sistema cuyos valores controla usted. Si el bucket está configurado como sitio web, a veces puede que quiera redirigir una solicitud de página a otra página o URL externa. En este caso, una página web será un objeto en su bucket. Amazon S3 almacena el valor de redirección de la página como metadatos del sistema cuyo valor controla usted.

Al crear objetos, puede configurar los valores de estos elementos de metadatos del sistema o actualizar los valores cuando lo necesite. Para obtener más información acerca de las clases de almacenamiento, consulte Uso de las clases de almacenamiento de Amazon S3.

Amazon S3 utiliza claves de AWS KMS para cifrar sus objetos de Amazon S3. AWS KMS cifra únicamente los datos del objeto. La suma de comprobación, junto con el algoritmo especificado, se almacenan como parte de los metadatos del objeto. Si se solicita el cifrado del lado del servidor para el objeto, la suma de comprobación se almacena de forma cifrada. Para obtener más información acerca del cifrado del lado del servidor, consulte Protección de los datos mediante el cifrado.

> **Note**
>
> El encabezado de las solicitudes PUT está limitado a un tamaño de 8 KB. En el encabezado de la solicitud PUT, los metadatos definidos por el sistema están limitados a un tamaño de 2 KB. El tamaño de los metadatos definidos por el sistema se mide con la suma del número de bytes de la codificación US-ASCII en cada clave y valor.

En la siguiente tabla se facilita una lista de metadatos definidos por el sistema y si puede actualizarlos.

Nombre	Descripción	¿El usuario puede modificar el valor?
`Date`	Fecha y hora actuales.	No
`Cache-Control`	Campo de encabezado general utilizado para especificar políticas de almacenamiento en caché.	Sí
`Content-Disposition`	Información de presentación de objetos.	Sí
`Content-Length`	Tamaño del objeto en bytes.	No

Nombre	Descripción	¿El usuario puede modificar el valor?
`Content-Type`	El tipo de objeto.	Sí
`Last-Modified`	Fecha de creación del objeto o última fecha de modificación, la que sea posterior. Para las cargas multiparte, la fecha de creación del objeto es la fecha de inicio de la carga multiparte.	No
`ETag`	Etiqueta de entidad (ETag) que representa una versión específica de un objeto. Para los objetos que no se cargan como una carga multiparte y que no se cifraron o se cifraron mediante el cifrado del lado del servidor con claves administradas de Amazon S3 (SSE-S3), ETag es un resumen MD5 de los datos.	No
`x-amz-server-side-encryption`	Encabezado que indica si se ha habilitado el cifrado del lado del servidor para el objeto y si dicho cifrado utiliza las claves de AWS Key Management Service (AWS KMS) (SSE-KMS) o claves de cifrado administrado por Amazon S3 (SSE-S3). Para obtener más información, consulte Protección de los datos con el cifrado del servidor.	Sí
`x-amz-checksum-crc32` , `x-amz-checksum-crc32c` , `x-amz-checksum-sha1` , `x-amz-checksum-sha256`	Encabezados que contienen la suma de comprobación o resumen del objeto. Como mucho, se configurará uno de estos encabezados a la vez, dependiendo del algoritmo de suma de comprobación que se indique a Amazon S3 que utilice. Para obtener más información sobre cómo elegir el algoritmo de suma de comprobación, consulte Comprobación de la integridad de objetos.	No

Nombre	Descripción	¿El usuario puede modificar el valor?
`x-amz-version-id`	Versión del objeto. Si activa el control de versiones en un bucket, Amazon S3 asigna un ID de versión a los objetos que se añades al bucket. Para obtener más información, consulte Usar el control de versiones en buckets de S3.	No
`x-amz-delete-marker`	Marcador booleano que indica si el objeto es un marcador de eliminación. Este marcador solo se usa en los buckets que tienen habilitado el control de versiones,	No
`x-amz-storage-class`	Clase de almacenamiento utilizada para almacenar el objeto. Para obtener más información, consulte Uso de las clases de almacenamiento de Amazon S3.	Sí
`x-amz-website-redirect-location`	Encabezado que redirige las solicitudes del objeto asociado hacia otro objeto del mismo bucket o hacia una URL externa. Para obtener más información, consulte (Opcional) Configuración del redireccionamiento de páginas web.	Sí
`x-amz-server-side-encryption-aws-kms-key-id`	Encabezado que indica el ID de la clave KMS de cifrado simétrico AWS KMS que se usó para cifrar el objeto. Este encabezado se usa solo cuando el encabezado `x-amz-server-side-encryption` está presente y tiene el valor `aws:kms`.	Sí
`x-amz-server-side-encryption-customer-algorithm`	Encabezado que indica si se ha habilitado el cifrado del lado del servidor con claves de cifrado proporcionadas por el cliente (SSE-C). Para obtener más información, consulte Uso de cifrado en el lado del servidor con claves proporcionadas por el cliente (SSE-C).	Sí

Nombre	Descripción	¿El usuario puede modificar el valor?
`x-amz-tagging`	El conjunto de etiquetas del objeto. El conjunto de etiquetas debe codificarse como parámetros de consulta de URL.	Sí

Metadatos de objetos definidos por el usuario

Al cargar un objeto, también puede asignar metadatos al objeto. Esta información opcional se facilita como par nombre-valor (clave-valor) al enviar una solicitud `PUT` o `POST` para crear el objeto. Cuando se cargan objetos con la API de REST, los nombres opcionales de metadatos definidos por el usuario deben comenzar con "`x-amz-meta-`" para distinguirlos de otros encabezados HTTP. Al recuperar el objeto con la API de REST, se devuelve este prefijo. Cuando carga objetos con la API SOAP, no es necesario el prefijo. Al recuperar el objeto con la API SOAP, el prefijo se elimina, independientemente de qué API se haya usado para cargar el objeto.

> **Note**
>
> La compatibilidad con SOAP por HTTP está obsoleta, pero SOAP aún se encuentra disponible con HTTPS. Las características nuevas de Amazon S3 no son compatibles con SOAP. En vez de usar SOAP, le recomendamos que utilice la API de REST o los SDK de AWS.

Cuando se recuperan metadatos con la API de REST, Amazon S3 combina encabezados que tengan el mismo nombre (sin tener en cuenta mayúsculas y minúsculas) en una lista delimitada por comas. Si algunos metadatos contienen caracteres no imprimibles, no se devuelven. En su lugar, se devuelve el encabezado `x-amz-missing-meta` con el número de entradas de metadatos no imprimibles como valor. La acción `HeadObject` recupera metadatos de un objeto sin devolver el objeto en sí. Esta operación es útil si solo está interesado en los metadatos de un objeto. Para utilizar `HEAD`, debe tener acceso `READ` al objeto. Para obtener más información, consulte el tema sobre el objeto Head en la referencia de la API de Amazon Simple Storage Service.

Los metadatos definidos por el usuario son un conjunto de pares clave-valor. Amazon S3 almacena las claves de metadatos definidos por el usuario en minúsculas.

Amazon S3 permite caracteres Unicode arbitrarios en los valores de metadatos.

Para evitar problemas en la presentación de estos valores de metadatos, debe cumplir con el uso de caracteres US-ASCII al utilizar REST y UTF-8 al utilizar SOAP o cargas basadas en navegador a través de POST.

Cuando se utilizan caracteres no US-ASCII en los valores de metadatos, la cadena Unicode proporcionada se examina en busca de caracteres no US-ASCII. Los valores de dichos encabezados se decodifican por caracteres según la RFC 2047 antes de almacenarse y codificarse según la RFC 2047 para que sean seguros para enviarlos por correo antes de devolverlos. Si la cadena contiene solo caracteres US-ASCII, se presenta tal cual.

A continuación, se muestra un ejemplo.

```
PUT /Key HTTP/1.1
Host: amzn-s3-demo-bucket1.s3.amazonaws.com
x-amz-meta-nonascii: ÄMÄZÕÑ S3

HEAD /Key HTTP/1.1
Host: amzn-s3-demo-bucket1.s3.amazonaws.com
x-amz-meta-nonascii: =?UTF-8?B?w4PChE3Dg8KEWsODwpXDg8KRIFMz?=

PUT /Key HTTP/1.1
Host: amzn-s3-demo-bucket1.s3.amazonaws.com
x-amz-meta-ascii: AMAZONS3

HEAD /Key HTTP/1.1
Host: amzn-s3-demo-bucket1.s3.amazonaws.com
x-amz-meta-ascii: AMAZONS3
```

> **Note**
>
> El encabezado de las solicitudes PUT está limitado a un tamaño de 8 KB. En el encabezado de la solicitud PUT, los metadatos definidos por el usuario están limitados a un tamaño de 2 KB. El tamaño de los metadatos definidos por el usuario se mide sumando el número de bytes de la codificación UTF-8 en cada clave y valor.

Si quiere obtener información para cambiar los metadatos del objeto después de que se haya cargado mediante la creación de una copia del objeto, la modificación y la sustitución del objeto anterior o la creación de una nueva versión, consulte Edición de metadatos de objeto en la consola de Amazon S3.

Edición de metadatos de objeto en la consola de Amazon S3

Puede utilizar la consola de Amazon S3 para editar metadatos de objetos de S3 existentes. Algunos metadatos son configurados por Amazon S3 cuando carga el objeto. Por ejemplo, `Content-Length` y `Last-Modified` son campos de metadatos de objetos definidos por el sistema que un usuario no puede modificar.

También puede configurar algunos metadatos cuando cargue el objeto, o después, puede editarlo a medida que cambien las necesidades. Por ejemplo, es posible que tenga un conjunto de objetos que almacene inicialmente en la clase de almacenamiento `STANDARD`. Con el tiempo, es posible que ya no necesite que estos datos estén altamente disponibles. Por lo tanto, cambia la clase de almacenamiento a `GLACIER` mediante la edición del valor de la clave `x-amz-storage-class` de `STANDARD` a `GLACIER`.

> **Note**
>
> Tenga en cuenta los siguientes problemas cuando edite metadatos de objeto en Amazon S3:
>
> - Esta acción crea una copia del objeto con la configuración actualizada y la fecha de última modificación. Si el control de versiones de S3 está activado, se crea una nueva versión del objeto y el objeto existente se convierte en una versión anterior. Si el control de versiones de S3 no está habilitado, una nueva copia del objeto reemplaza al objeto original. La Cuenta de AWS con el rol de IAM que cambia la propiedad también se convierte en la propietaria del nuevo objeto o (versión del objeto).
> - Para utilizar la consola de Amazon S3 para editar metadatos de un objeto que tiene etiquetas definidas por el usuario, debe tener también el permiso `s3:GetObjectTagging`. Si va a utilizar la consola de Amazon S3 para editar los metadatos de un objeto que no tiene etiquetas definidas por el usuario pero que tiene un tamaño superior a 16 MB, también debe tener el permiso `s3:GetObjectTagging`.
>
>   Si la política de bucket de destino deniega la acción `s3:GetObjectTagging`, los metadatos del objeto se actualizarán, pero las etiquetas definidas por el usuario se eliminarán del objeto y aparecerá un error.

- La edición de metadatos actualiza los valores de los nombres de clave existentes.
- Los objetos cifrados con claves de cifrado proporcionadas por el cliente (SSE-C) no se pueden copiar con la consola. Debe utilizar la AWS CLI, el SDK de AWS o la API de REST de Amazon S3.

Warning

Cuando edite metadatos de carpetas, espere a que finalice la operación `Edit metadata` antes de agregar nuevos objetos a la carpeta. De lo contrario, también se podrían editar objetos nuevos.

En los temas siguientes se describe cómo editar metadatos de un objeto mediante la consola de Amazon S3.

Edición de metadatos definidos por el sistema

Puede configurar algunos metadatos del sistema para un objeto de S3, pero no todos. Para obtener una lista de metadatos definidos por el sistema y saber si puede modificar sus valores, consulte Metadatos de objetos definidos por el sistema.

Para editar metadatos definidos por el sistema de un objeto, realice las siguientes acciones:

1. Inicie sesión en la AWS Management Console y abra la consola de Amazon S3 en https://console.aws.amazon.com/s3/.
2. Desplácese hasta el bucket o carpeta de Amazon S3 y seleccione la casilla de verificación situada a la izquierda de los nombres de los objetos con los metadatos que desea editar.
3. En el menú Actions (Acciones), elija Edit actions (Editar acciones) y elija Edit metadata (Editar metadatos).
4. Revise los objetos descritos y elija Add metadata (Agregar metadatos).
5. Para el metadato Type (Tipo), seleccione System-defined (Definidos por el sistema).
6. Especifique una Key (Clave) única y el metadato Value (Valor).
7. Para editar metadatos adicionales, elija Add metadata (Añadir metadatos). También puede elegir Remove (Eliminar) para eliminar un conjunto de valores de clave de tipo.

8. Cuando haya terminado, elija Edit metadata (Editar metadatos) y Amazon S3 editará los metadatos de los objetos especificados.

Edición de metadatos definidos por el usuario

Puede editar metadatos definidos por el usuario de un objeto combinando el prefijo de metadatos, `x-amz-meta-` y un nombre que elija para crear una clave personalizada. Por ejemplo, si añade el nombre personalizado `alt-name`, la clave de los metadatos será `x-amz-meta-alt-name`.

Los metadatos definidos por el usuario pueden tener un tamaño total de hasta 2 KB. Para calcular el tamaño total de los metadatos definidos por el usuario, sume el número de bytes en la codificación UTF-8 para cada clave y valor. Tanto las claves como sus valores deben cumplir los estándares del American Standard Code for Information Interchange (ASCII, Código Estándar Estadounidense para el Intercambio de Información) de los EE. UU. Para obtener más información, consulte Metadatos de objetos definidos por el usuario.

Para editar metadatos definidos por el usuario de un objeto, realice las siguientes acciones:

1. Inicie sesión en la AWS Management Console y abra la consola de Amazon S3 en https://console.aws.amazon.com/s3/.
2. En la lista Buckets, elija el nombre del bucket que contiene los objetos a los que desea agregar metadatos.

   Si lo desea, también puede ir una carpeta.
3. En la lista Objects (Objetos), seleccione la casilla de verificación situada junto a los nombres de los objetos a los que desea agregar metadatos.
4. En el menú Actions (Acciones), elija Edit metadata (Editar metadatos).
5. Revise los objetos descritos y elija Add metadata (Agregar metadatos).
6. Para el metadato Type (Tipo), elija User-defined (Definidos por el usuario).
7. Escriba una Key (Clave) única y personalizada después de `x-amz-meta-`. Introduzca también un metadato Value (Valor).
8. Para añadir metadatos adicionales, elija Add metadata (Añadir metadatos). También puede elegir Remove (Eliminar) para eliminar un conjunto de valores de clave de tipo.
9. Elija Edit medatada (Editar metadatos).

   Amazon S3 edita los metadatos de los objetos especificados.

Carga de objetos

Al cargar un archivo en Amazon S3, se guarda como un objeto de S3. Los objetos constan de los datos y metadatos del archivo que describen el objeto. Puede haber un número ilimitado de objetos en un bucket. Para poder cargar archivos en un bucket de Amazon S3, debe escribir permisos para el bucket. Para obtener más información acerca de los permisos de acceso, consulte Administración de identidades y accesos para Amazon S3.

Puede cargar cualquier tipo de archivo, como imágenes, copias de seguridad, datos, películas, etc., en un bucket de S3. El tamaño máximo de un archivo que puede cargar con la consola de Amazon S3 es de 160 GB. Para cargar un archivo que sobrepase los 160 GB de tamaño, utilice la AWS Command Line Interface (AWS CLI), el SDK de AWS o la API de REST de Amazon S3.

Si carga un objeto con un nombre de clave que ya existe en un bucket con el control de versiones activado, Amazon S3 crea otra versión del objeto en vez de reemplazar el objeto existente. Para obtener más información sobre el control de versiones, consulte Uso de la consola de S3.

En función del tamaño de los datos que cargue, Amazon S3 ofrece las siguientes opciones:

- Cargar un objeto en una única operación mediante los SDK de AWS, la API de REST o la AWS CLI: con una única operación PUT, se puede cargar un único objeto de hasta 5 GB.
- Cargar un solo objeto mediante la consola de Amazon S3: con la consola de Amazon S3, se puede cargar un único objeto de hasta 160 GB.
- Cargar un objeto en partes mediante los SDK de AWS, la API de REST o la AWS CLI: con la API de carga multiparte, se puede cargar un solo objeto grande, de hasta 5 TB.

  La operación de la API de carga multiparte está diseñada para mejorar la experiencia de carga para objetos más grandes. Puede cargar un objeto en partes. Estas partes de objetos se pueden cargar independientemente, en cualquier orden y en paralelo. Puede usar la carga multiparte para objetos que ocupen de 5 MB a 5 TB. Para obtener más información, consulte Carga y copia de objetos con la carga multiparte.

Cuando se carga un objeto, el objeto se cifra automáticamente mediante cifrado del lado del servidor con claves administradas de Amazon S3 (SSE-S3) de forma predeterminada. Al descargarlo, el objeto se descifra. Para obtener más información, consulte Establecer el comportamiento del cifrado predeterminado del lado del servidor para los buckets de Amazon S3 y Protección de los datos mediante el cifrado.

Al cargar un objeto, si quiere utilizar un tipo de cifrado predeterminado diferente, también puede especificar el cifrado del lado del servidor con claves de AWS Key Management Service (AWS KMS) (SSE-KMS) en sus solicitudes PUT de S3 o definir la configuración de cifrado predeterminada en el bucket de destino para usar SSE-KMS para cifrar los datos. Para obtener más información sobre SSE-KMS, consulte Especificación del cifrado del lado del servidor con AWS KMS (SSE-KMS). Si desea utilizar una clave de KMS propiedad de una cuenta diferente, primero debe tener permiso para utilizar la clave. Para obtener más información sobre los permisos entre cuentas para las claves de KMS, consulte Crear claves de KMS que otras cuentas puedan utilizar en la Guía para desarrolladores de AWS Key Management Service.

Si aparece un error Access Denied (403 Forbidden) en Amazon S3, consulte Solucionar errores de acceso denegado (403 Prohibido) en Amazon S3 para obtener más información sobre las causas comunes de dicho error.

Uso de la consola de S3

Este procedimiento explica cómo cargar objetos y carpetas a un bucket de Amazon S3 mediante la consola.

Cuando carga un objeto, el nombre de la clave de objeto es el nombre del archivo y los prefijos opcionales. En la consola de Amazon S3, puede crear carpetas para organizar sus objetos. En Amazon S3, las carpetas se representan como prefijos que aparecen en el nombre de la clave de objeto. Si carga un objeto individual en una carpeta de la consola de Amazon S3, el nombre de la carpeta se incluirá en el nombre de la clave de objeto.

Por ejemplo, si carga un objeto denominado `sample1.jpg` en una carpeta denominada `backup`, el nombre de la clave es `backup/sample1.jpg`. Sin embargo, el objeto se mostrará en la consola como `sample1.jpg` la carpeta `backup`. Para obtener más información sobre nombres de clave, consulte Trabajar con metadatos de objeto.

> **Note**
>
> Si cambia el nombre de un objeto o cambia cualquiera de las propiedades de la consola de Amazon S3 (por ejemplo, la clase de almacenamiento, el cifrado o los metadatos), se crea un nuevo objeto para reemplazar el anterior. Si el control de versiones de S3 está activado, se crea una nueva versión del objeto y el objeto existente se convierte en una versión anterior. El rol que cambia la propiedad también se convierte en el propietario del nuevo objeto o (versión del objeto).

Cuando carga una carpeta, Amazon S3 carga todos los archivos y subcarpetas de la carpeta especificada en su bucket. Luego asigna un nombre de clave de objeto, que es una combinación del nombre del archivo cargado y el nombre de la carpeta. Por ejemplo, si carga una carpeta denominada `/images` que contiene dos archivos, `sample1.jpg` y `sample2.jpg`, Amazon S3 carga los archivos y les asigna los nombres de clave correspondientes, `images/sample1.jpg` e `images/sample2.jpg`. Los nombres de clave incluyen el nombre de la carpeta como un prefijo. La consola de Amazon S3 muestra solo la parte del nombre de clave siguiente a la última `/`. Por ejemplo, dentro de una carpeta `images`, los objetos `images/sample1.jpg` y `images/sample2.jpg` se muestran como `sample1.jpg` y `sample2.jpg`.

Para cargar carpetas y archivos a un bucket de S3

1. Inicie sesión AWS Management Console Management Console y abra la consola de Amazon S3 en https://console.aws.amazon.com/s3/.
2. En el panel de navegación izquierdo, elija Instancias.
3. En la lista Buckets (Buckets), elija el nombre del bucket en el que desea cargar sus carpetas o archivos.
4. Seleccione Upload.
5. En la ventana Upload (Cargar), realice una de las siguientes acciones:
   - Arrastre y suelte archivos y carpetas en la ventana Upload (Cargar).
   - Elija Agregar archivo o Agregar carpeta, elija los archivos o carpetas que desee cargar y Abrir.
6. Para habilitar el control de versiones, en Destination (Destino), seleccione Enable Bucket Versioning (Activar control de versiones de bucket).
7. Para cargar los archivos y carpetas enumerados sin configurar opciones de carga adicionales, en la parte inferior de la página, seleccione Upload (Cargar).

   Amazon S3 carga sus objetos y carpetas. Cuando finalice la carga, puede ver un mensaje de éxito en la página Cargar: estado.

Para configurar propiedades de objeto adicionales

1. Para cambiar los permisos de la lista de control de acceso, elija Permissions (Permisos).
2. En Access control list (ACL) (Lista de control de acceso [ACL]), edite los permisos.

Para obtener información acerca de los permisos de acceso a objetos, consulte Uso de la consola S3 para establecer permisos de ACL para un objeto. Puede conceder acceso de lectura a los objetos al público (a todo el mundo) para todos los archivos que esté cargando. Sin embargo, recomendamos no cambiar la configuración predeterminada para el acceso de lectura pública. Otorgar acceso de lectura público es aplicable a un pequeño conjunto de casos de uso, por ejemplo cuando los buckets se utilizan para sitios web. Siempre puede cambiar los permisos del objeto después de cargarlo.

3. Para configurar otras propiedades, elija Properties (Propiedades).
4. En la sección Clase de almacenamiento, elija la clase de almacenamiento para los archivos que esté cargando.

   Para obtener más información acerca de las clases de almacenamiento, consulte Uso de las clases de almacenamiento de Amazon S3.
5. Para actualizar la configuración de cifrado de los objetos, en Server-side encryption settings (Configuración de cifrado del lado del servidor), haga lo siguiente.

   a. Elija Specify an encryption key (Especificar una clave de cifrado).

   b. En Configuración del cifrado, elija Usar la configuración del bucket para el cifrado predeterminado o Anular la configuración del bucket para el cifrado predeterminado.

   c. Si elige Anular la configuración del bucket para el cifrado predeterminado, debe configurar los siguientes ajustes de cifrado.

      - Para cifrar los archivos cargados con claves que administra Amazon S3, seleccione Clave administrada de Amazon S3 (SSE-S3).

        Para obtener más información, consulte Uso del cifrado del servidor con claves administradas por Amazon S3 (SSE-S3).
      - Si desea cifrar los archivos cargados mediante claves almacenadas en AWS Key Management Service (AWS KMS), elija la Clave de AWS Key Management Service (SSE-KMS). A continuación, elija una de las siguientes opciones para la clave de AWS KMS:
        - Para seleccionar en una lista de claves de KMS disponibles, marque Elija entre sus claves de AWS KMS keys y seleccione su clave de KMS en la lista de claves disponibles.

          En esta lista aparecen tanto la Clave administrada de AWS (`aws/s3`) como las claves administradas por el cliente. Para obtener más información acerca de las claves

administradas por el cliente, consulte Claves de cliente y claves de AWS en la Guía para desarrolladores de AWS Key Management Service.

- Para introducir el ARN de la clave de KMS, elija Introducir el ARN de la clave de AWS KMS key e introduzca el ARN de la clave de KMS en el campo que aparece.
- Para crear una nueva clave administrada por el cliente en la consola de AWS KMS, elija Crear una clave de KMS.

  Para obtener más información acerca de cómo crear una AWS KMS key, consulte Creación de claves en la AWS Key Management Service Guía para desarrolladores.

> **Important**
>
> Solo puede utilizar las claves de KMS que estén disponibles en la misma Región de AWS del bucket. La consola de Amazon S3 solo muestra las primeras 100 claves de KMS de la misma región del bucket. Para utilizar una clave de KMS que no aparezca en la lista, debe introducir el ARN de la clave de KMS. Si desea utilizar una clave de KMS propiedad de una cuenta de diferente, primero debe tener permiso para utilizar la clave y, después, debe introducir el ARN de la clave de KMS.
>
> Amazon S3 admite solo claves KMS de cifrado simétricas y no claves KMS asimétricas. Para obtener más información, consulte Identificación de claves de KMS simétricas y asimétricas en la Guía para desarrolladores de AWS Key Management Service.

6. Para utilizar sumas de comprobación adicionales, elija On (Activado). A continuación, en Checksum function (Función de suma de comprobación), elija la función que desearía utilizar. Amazon S3 calcula y almacena el valor de la suma de comprobación después de recibir todo el objeto. Puede utilizar el recuadro Precalculated value (Valor precalculado) para proporcionar un valor precalculado. Si lo hace, Amazon S3 compara el valor que proporcionó con el valor que calcula. Si los valores no coinciden, Amazon S3 genera un error.

   Las sumas de comprobación adicionales le permiten especificar el algoritmo de suma de comprobación que desea utilizar para verificar los datos. Para obtener más información sobre sumas de comprobación adicionales, consulte Comprobación de la integridad de objetos.
7. Para agregar etiquetas a todos los objetos que va a cargar, elija Add tag (Agregar etiqueta). Escriba un nombre de etiqueta en el campo Clave. Ingrese un valor para la etiqueta.

El etiquetado de objetos le permite categorizar el almacenamiento. Cada etiqueta es un par clave-valor. Los valores de clave y de etiqueta distinguen entre mayúsculas y minúsculas. Puede tener hasta 10 etiquetas por cada objeto. Una clave de etiqueta puede tener una longitud de hasta 128 caracteres Unicode y los valores de etiqueta pueden tener una longitud de hasta 255 caracteres Unicode. Para obtener más información acerca de las etiquetas de objeto, consulte Categorización del almacenamiento mediante etiquetas.

8. Para agregar metadatos, elija Add metadata (Agregar metadatos).

   a. En Type (Tipo), elija System defined (Definido por sistema) o User defined (Definido por el usuario).

      Para los metadatos definidos por el sistema, puede seleccionar encabezados HTTP comunes, como Content-Type (Tipo de contenido) y Content-Disposition (Disposición de contenido). Para obtener una lista de metadatos definidos por el sistema e información sobre si puede agregar el valor, consulte Metadatos de objetos definidos por el sistema. Cualquier metadato que comience con el prefijo `x-amz-meta-` se trata como metadato definido por el usuario. Los metadatos definidos por el usuario se almacenan con el objeto y se devuelven cuando lo descarga. Tanto las claves como sus valores deben cumplir los estándares ASCII de EE. UU. Los metadatos definidos por el usuario pueden tener un tamaño de hasta 2 KB. Para obtener más información acerca de los metadatos definidos por el sistema y por el usuario, consulte Trabajar con metadatos de objeto.

   b. Para Key (Clave), elija una clave.

   c. Escriba un valor para la clave.

9. Para cargar los objetos, selecciona Upload (Cargar).

   Amazon S3 carga su objeto. Cuando finalice la carga, puede ver un mensaje de éxito en la página Upload: status (Cargar: estado).

10. Seleccione Exit (Salir).

Uso de los AWS SDK

Puede utilizar los SDK de AWS para cargar objetos en Amazon S3. Los SDK proporcionan bibliotecas de encapsulamiento para que pueda cargar datos fácilmente. Para obtener información, consulte la Lista de SDK admitidos.

Aquí se muestran algunos ejemplos con algunos SDK seleccionados:

.NET

El siguiente código de C# crea dos objetos con dos solicitudes de `PutObjectRequest`:

- La primera solicitud de `PutObjectRequest` guarda una cadena de texto como muestra de datos del objeto. También especifica los nombres del bucket y de la clave del objeto.
- La segunda solicitud de `PutObjectRequest` carga un archivo especificando el nombre del archivo. Esto requiere también especificar el encabezado de `ContentType` y los metadatos opcionales del objeto (un título).

Para obtener información acerca de cómo configurar y ejecutar ejemplos de código, consulte Introducción al SDK de AWS para .NET en la Guía para desarrolladores del SDK de AWS para .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

En el siguiente ejemplo se crean dos objetos. El primer objeto tiene una cadena de texto como datos y el segundo objeto es un archivo. El ejemplo crea el primer objeto especificando el nombre de bucket, la clave de objeto y los datos de texto directamente en una llamada a `AmazonS3Client.putObject()`. El ejemplo crea el segundo objeto utilizando un `PutObjectRequest` que especifica el nombre de bucket, la clave de objeto y la ruta del archivo. El `PutObjectRequest` también especifica el encabezado de `ContentType` y los metadatos del título.

Para obtener instrucciones sobre cómo crear y probar una muestra funcional, consulte Introducción en la Guía del desarrollador de AWS SDK for Java.

```
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

En el ejemplo siguiente se carga un archivo existente en un bucket de Amazon S3 en una región específica.

```
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

En este tema se detalla el proceso de uso de las clases de AWS SDK for PHP para cargar un objeto de hasta 5 GB de tamaño. Para archivos más grandes, debe utilizar la operación de la API de carga multiparte. Para obtener más información, consulte Carga y copia de objetos con la carga multiparte.

Para obtener más información acerca de la API del SDK de AWS para Ruby, consulte SDK de AWS para Ruby, versión 2.

Example — Creación de un objeto en un bucket de Amazon S3 mediante la carga de datos

En el siguiente ejemplo en PHP se crea un objeto en un bucket específico cargando datos con el método `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

AWS SDK for Ruby, versión 3, tiene dos maneras de cargar un objeto en Amazon S3. La primera usa un cargador de archivos administrado, que facilita la carga de archivos de cualquier tamaño desde el disco. Para utilizar el método del cargador de archivos administrado:

1. Cree una instancia de la clase `Aws::S3::Resource`.
2. Haga referencia al objeto objetivo con el nombre del bucket y la clave. Los objetos residen en un bucket y tienen claves únicas que identifica cada objeto.
3. Llame a `#upload_file` en el objeto.

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 segunda manera en la que la versión 3 de AWS SDK for Ruby puede cargar un objeto es usar el método `#put` de `Aws::S3::Object`. Esto resulta útil si el objeto es una cadena o un objeto I/O que no sea un archivo en el disco. Para usar este método:

1. Cree una instancia de la clase `Aws::S3::Resource`.
2. Haga referencia al objeto objetivo con el nombre del bucket y la clave.
3. Llame a `#put`, pasando la cadena o el objeto 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__
```

Uso de la API de REST

Puede enviar solicitudes REST para cargar un objeto. Puede enviar una solicitud PUT para cargar datos en una única operación. Para obtener más información, consulte PUT Object.

Mediante AWS CLI

Envíe una solicitud PUT para cargar un objeto de hasta 5 GB en una única operación. Para obtener más información, consulte el ejemplo de `PutObject` en la referencia de comandos de AWS CLI.

Carga y copia de objetos con la carga multiparte

La carga multiparte permite cargar un solo objeto como un conjunto de partes. Cada parte es una parte contigua de los datos del objeto. Puede cargar estas partes del objeto de forma independiente y en cualquier orden. Si la transmisión de cualquier parte falla, puede retransmitir esta parte sin que las demás partes se vean afectadas. Después de cargar todas las partes del objeto, Amazon S3 las combina y crea el objeto. Por lo general, cuando el tamaño del objeto alcanza los 100 MB, deberá usar las cargas multipartes en lugar de cargar el objeto en una única operación.

El uso de la carga multiparte proporciona las siguientes ventajas:

- Mayor rendimiento: puede cargar las partes al mismo tiempo para aumentar el rendimiento.
- Recuperación rápida ante cualquier problema de red: una parte de tamaño más pequeño reduce el impacto de tener que reiniciar una carga fallida debido a un error de red.
- Detención y reanudación de cargas de objetos: puede cargar las partes del objeto con el paso del tiempo. Después de que inicia una carga multiparte, no hay fecha de caducidad, por lo tanto, debe completar o detener de forma explícita la carga multiparte.
- Inicio de una carga antes de conocer el tamaño final del objeto: puede cargar un objeto a medida que lo crea.

Le recomendamos que use la carga multiparte de las siguientes maneras:

- Si carga objetos grandes en una red estable de banda ancha, use la carga multiparte para aumentar al máximo el uso de su ancho de banda disponible cargando los objetos en partes y en paralelo para un rendimiento en varios subprocesos.
- Si realiza la carga en una red irregular, use la carga multiparte para aumentar la resiliencia ante errores de red evitando reinicios de la carga. Al usar la carga multiparte, debe volver a intentar cargar solo las partes que se han interrumpido durante la carga. No necesita reiniciar la carga de su objeto desde el principio.

> **Note**
>
> Para obtener más información sobre el uso de la clase de almacenamiento Amazon S3 Express One Zone con buckets de directorio, consulte ¿Qué es S3 Express One Zone? y Buckets de directorio. Para obtener más información sobre el uso de una carga multiparte

con S3 Express One Zone y buckets de directorio, consulte Uso de las cargas multiparte con buckets de directorio.

Proceso de carga multiparte

La carga multiparte es un proceso de tres pasos: se inicia la carga, se cargan las partes del objeto y, después de haber cargado todas las partes, se completa la carga multiparte. Al recibir la solicitud de carga multiparte completa, Amazon S3 crea el objeto a partir de las partes cargadas para que pueda obtener acceso al objeto como lo haría con cualquier otro objeto de su bucket.

Puede mostrar todas las cargas multipartes en curso u obtener una lista de las partes que ha cargado en una carga multiparte específica. En esta sección, se explicarán cada una de estas operaciones.

Inicio de la carga multiparte

Al enviar una solicitud para iniciar una carga multiparte, Amazon S3 devuelve una respuesta con un ID de carga, que es un identificador único para su carga multiparte. Debe incluir este ID de carga siempre que cargue partes, muestre partes, complete una carga o pare una carga. Si desea proporcionar metadatos que describen el objeto que está cargando, debe proporcionarlos en la solicitud para iniciar la carga multiparte.

Carga de partes

Al cargar una parte, además del ID de carga, debe especificar un número de parte. Puede seleccionar cualquier número de parte comprendido entre 1 y 10 000. Un número de parte identifica exclusivamente una parte y su posición en el objeto que se está cargando. El número de parte que elija no tiene que ser necesariamente una secuencia consecutiva (por ejemplo: puede ser 1, 5 y 14). Si carga una parte nueva con el mismo número que una parte ya cargada, se sobrescribirá la parte existente.

Cuando cargue una parte, Amazon S3 devolverá una etiqueta de entidad (ETag) para la parte como encabezado en la respuesta. Para cada carga de parte, debe anotar el número de parte y el valor de ETag. Debe incluir estos valores en la solicitud posterior para completar la carga multiparte. Cada parte tendrá su propia ETag en el momento de la carga. Sin embargo, una vez que se complete la carga de varias partes y todas se hayan consolidado, todas las partes estarán agrupadas en una ETag como suma de comprobación de las sumas de comprobación.

> **Note**
>
> Después de iniciar una carga multiparte y cargar una o más partes, debe completar o parar la carga multiparte para que no le cobren por el almacenamiento de las partes cargadas. Solo después de completar o parar una carga multiparte, Amazon S3 liberará el almacenamiento de las partes y parará el cobro del almacenamiento de partes
> Después de parar una carga multiparte, no puede volver a cargar ninguna parte con ese ID de carga. Si la carga de alguna de las partes estuviera en curso, todavía se puede ejecutar correctamente o producir un error una vez detenida. Para asegurarse de que se libera todo el espacio de almacenamiento consumido por las partes, debe parar una carga multiparte solo después de haber completado las cargas de todas las partes.

Finalización de la carga multiparte

Al completar una carga multiparte, Amazon S3 crea un objeto al concatenar las partes en orden ascendente según el número de parte. Si se proporcionaron los metadatos de algún objeto en la solicitud inicio de carga multiparte, Amazon S3 asocia estos metadatos al objeto. Después de una solicitud de completar realizada correctamente, las partes ya no existirán.

La solicitud carga multiparte completa debe incluir el ID de carga y una lista de ambos números de parte y valores correspondientes de ETag. La respuesta de Amazon S3 incluye una ETag que identifica de forma exclusiva los datos de objetos combinados. Esta ETag no es necesariamente un hash de MD5 de los datos del objeto.

Ejemplo de llamadas de carga multiparte

En este ejemplo, suponga que está generando una carga multiparte para un archivo de 100 GB. En tal caso tendría las siguientes llamadas a la API para todo el proceso. Habría un total de 1002 llamadas a la API.

- Una llamada `CreateMultipartUpload` para iniciar el proceso.
- 1000 llamadas `UploadPart` individuales, cada una cargando una parte de 100 MB, con un tamaño total de 100 GB.
- Una llamada `CompleteMultipartUpload` para finalizar el proceso.

Listas de cargas multiparte

Puede enumerar las partes de una carga multiparte específica o todas las cargas multipartes en curso. La operación de lista de partes devuelve la información de las partes que ha cargado para una carga multiparte específica. Para cada solicitud de lista de partes, Amazon S3 devuelve la información de las partes para la carga multiparte específica, hasta un máximo de 1 000 partes. Si hay más de 1 000 partes en la carga multiparte, debe enviar una serie de solicitudes de lista de partes para recuperar todas las partes. Tenga en cuenta que la lista de partes que se devuelve no incluye las partes que no hayan terminado de cargarse. Con la operación lista de cargas multiparte, puede obtener una lista de las cargas multiparte en curso.

Una carga multiparte en curso es una carga iniciada, pero que aún no se ha completado ni parado. Cada solicitud devuelve 1 000 cargas multipartes como máximo. Si hay más de 1 000 cargas multiparte en curso, debe enviar otras solicitudes para recuperar las cargas multiparte restantes. Solo utilice la lista devuelta para fines de verificación. No utilice el resultado de esta lista al enviar una solicitud para completar la carga multiparte. En cambio, mantenga su propia lista de números de parte que especificó al cargarlas y los valores correspondientes de ETag que devuelve Amazon S3.

Sumas de comprobación con operaciones de carga multiparte

Al cargar un objeto en Amazon S3, puede especificar un algoritmo de suma de comprobación para que lo utilice Amazon S3. Amazon S3 utiliza MD5 de forma predeterminada para verificar la integridad de los datos; sin embargo, puede especificar un algoritmo de suma de comprobación adicional para utilizarlo. Al utilizar MD5, Amazon S3 calcula la suma de comprobación de todo el objeto multiparte una vez finalizada la carga. Esta suma de comprobación no es de todo el objeto, sino más bien una suma de comprobación de las sumas de comprobación de cada parte individual.

Cuando indica a Amazon S3 que utilice sumas de comprobación adicionales, Amazon S3 calcula el valor de la suma de comprobación de cada parte y almacena los valores. Puede utilizar la API o el SDK para recuperar el valor de la suma de comprobación de partes individuales mediante `GetObject` o `HeadObject`. Si quiere recuperar los valores de la suma de comprobación de partes individuales de las cargas multiparte que aún están en proceso, puede utilizar `ListParts`.

> **Important**
>
> Si utiliza una carga multiparte con sumas de comprobación adicionales, los números de parte multiparte deben ser consecutivos. Al utilizar sumas de comprobación adicionales, si intenta completar una solicitud de carga multiparte con números de parte no consecutivos, Amazon S3 genera un error `500 Internal Server Error` de HTTP.

Para obtener más información sobre cómo funcionan las sumas de comprobación con objetos multiparte, consulte Comprobación de la integridad de objetos.

Operaciones de carga multiparte simultáneas

En un entorno de desarrollo distribuido, es posible que la aplicación inicie varias actualizaciones en el mismo objeto simultáneamente. La aplicación puede iniciar varias cargas multipartes con la misma clave de objeto. Para cada una de estas cargas, la aplicación puede cargar las partes y enviar una solicitud de carga completa a Amazon S3 para crear el objeto. Cuando los buckets tienen el control de versiones de S3 habilitado, siempre se creará una nueva versión cuando se complete una carga multiparte. Al iniciar varias cargas multiparte que utilizan la misma clave de objeto en un bucket con el control de versiones habilitado, la versión actual del objeto viene determinada por la carga que se haya iniciado más recientemente (`createdDate`). Por ejemplo, supongamos que inicia una solicitud CreateMultipartUpload para un objeto a las 10:00 h. A continuación, envía una segunda solicitud `CreateMultipartUpload` para el mismo objeto a las 11:00 h. Como la segunda solicitud se envió más recientemente, el objeto cargado por la solicitud de las 11:00 h será la versión actual, incluso si la primera carga se completa después de la segunda. Para buckets que no tienen el control de versiones habilitado, es posible que tenga prioridad alguna otra solicitud recibida entre el momento en que se inicia y cuando se completa una carga multiparte.

> **Note**
>
> Es posible que tenga prioridad alguna otra solicitud recibida entre el momento en que inició y completó la carga multiparte. Por ejemplo, si otra operación elimina una clave luego de que usted inicia una carga multiparte con esa clave antes de completarla, la respuesta de carga multiparte completa podría indicar una creación correcta del objeto sin que usted vea el objeto.

Carga multiparte y precios

Después de iniciar una carga multiparte, Amazon S3 retiene todas las partes hasta que complete o detenga la carga. Durante la vida útil, se le cobrará por todo el almacenamiento, el ancho de banda y las solicitudes para esta carga multiparte y sus partes asociadas.

Estas partes se cobran según la clase de almacenamiento especificada cuando se cargaron las partes. Una excepción a esto son las partes cargadas en S3 Glacier Flexible Retrieval o S3 Glacier Deep Archive. Las partes multipartes en curso para PUT a la clase de almacenamiento

S3 Glacier Flexible Retrieval se facturan como S3 Glacier Flexible Retrieval Staging Storage a las tarifas de almacenamiento de S3 Standard hasta que se complete la carga. Además, tanto CreateMultipartUpload como UploadPart se facturan según las tarifas de S3 Standard. Solo la solicitud CompleteMultipartUpload se factura con la tarifa de S3 Glacier Flexible Retrieval. Del mismo modo, las partes multipartes en curso para PUT a la clase de almacenamiento S3 Glacier Deep Archive se facturan como S3 Glacier Flexible Retrieval Staging Storage a las tarifas de almacenamiento de S3 Standard hasta que se complete la carga, con solo la solicitud CompleteMultipartUpload cobrada según las tarifas de S3 Glacier Deep Archive.

No se le cobrará por estos elementos, si para la carga multiparte, ya que Amazon S3 elimina los artefactos cargados y cualquier parte que haya cargado. No se cobran gastos de eliminación anticipada por eliminar cargas multiparte incompletas, independientemente de la clase de almacenamiento especificada. Para obtener más información acerca de los precios, consulte Precios de Amazon S3.

> **Note**
>
> Para reducir los costos de almacenamiento, se recomienda configurar una regla del ciclo de vida para eliminar las cargas multiparte incompletas al cabo de un número específico de días mediante la acción `AbortIncompleteMultipartUpload`. Para obtener más información sobre cómo crear una regla de ciclo de vida para eliminar las cargas multiparte incompletas, consulte Configuración de una política de ciclo de vida del bucket para eliminar cargas multiparte incompletas.

Compatibilidad de la API con las cargas multiparte

Estas bibliotecas proporcionan una abstracción de alto nivel que facilita la carga multiparte de objetos. Sin embargo, si la aplicación lo requiere, puede utilizar directamente la API de REST. Las siguientes secciones de la referencia de API de Amazon Simple Storage Service describen la API de REST para la carga multiparte.

Para ver un tutorial de carga de varias partes donde se utilicen funciones de AWS Lambda, consulte Uploading large objects to Amazon S3 using multipart upload and transfer acceleration.

- Crear carga multiparte
- Upload Part
- Upload Part (Copy)

- Complete Multipart Upload
- Abort Multipart Upload
- List Parts
- List Multipart Uploads

Compatibilidad de AWS Command Line Interface con cargas multiparte

En los siguientes temas de AWS Command Line Interface, se describen las operaciones para la carga multiparte.

- Initiate Multipart Upload
- Upload Part
- Upload Part (Copy)
- Complete Multipart Upload
- Abort Multipart Upload
- List Parts
- List Multipart Uploads

Compatibilidad de AWS SDK con cargas multiparte

Puede utilizar un AWS SDK para cargar un objeto en partes. Para obtener una lista de los AWS SDK compatibles con la acción de la API, consulte:

- Crear carga multiparte
- Upload Part
- Upload Part (Copy)
- Complete Multipart Upload
- Abort Multipart Upload
- List Parts
- List Multipart Uploads

API y permisos de carga multiparte

Debe tener los permisos necesarios para utilizar las operaciones de carga multiparte. Puede utilizar listas de control de acceso (ACL), la política de bucket o la política de usuario para conceder permisos para realizar estas operaciones. En la siguiente tabla se muestran los permisos requeridos para varias operaciones de carga multiparte al utilizar las ACL, una política de bucket o una política de usuario.

Acción	Permisos necesarios
Crear carga multiparte	Debe tener permiso para realizar la acción `s3:PutObject` en un objeto a fin de crear la carga multiparte. El propietario del bucket puede permitir a otros clientes realizar la acción `s3:PutObject` .
Initiate Multipart Upload	Debe tener permiso para realizar la acción `s3:PutObject` en un objeto para iniciar la carga multiparte. El propietario del bucket puede permitir a otros clientes realizar la acción `s3:PutObject` .
Initiator	Elemento contenedor que identifica quién inició la carga multiparte. Si el iniciador es una Cuenta de AWS, este elemento proporcionará la misma información que el elemento Propietario. Si el iniciador es un usuario de IAM, este elemento proporciona el ARN de usuario y el nombre para mostrar.
Upload Part	Debe tener permiso para realizar la acción `s3:PutObject` en un objeto para cargar una parte. El propietario del bucket debe permitir al iniciador realizar la acción `s3:PutObject` en un objeto para que cargue una parte de ese objeto.
Upload Part (Copy)	Debe tener permiso para realizar la acción `s3:PutObject` en un objeto para cargar una parte. Dado que está cargando una parte de un objeto existente, debe tener permiso para realizar la acción `s3:GetObject` en el objeto de origen. Para que el iniciador cargue una parte del objeto, el propietario del bucket debe permitir al iniciador realizar la acción `s3:PutObject` en un objeto.

Acción	Permisos necesarios
Complete Multipart Upload	Debe tener permiso para realizar la acción `s3:PutObject` en un objeto para completar una carga multiparte. El propietario del bucket debe permitir al iniciador realizar la acción `s3:PutObject` en un objeto para que complete una carga multiparte para ese objeto.
Detener carga multiparte	Debe tener permiso para realizar la acción `s3:AbortMultipartUpload` para parar una carga multiparte. De forma predeterminada, el propietario del bucket y el iniciador de la carga multiparte deben tener permiso para realizar esta acción como parte de las políticas de bucket e IAM. Si el iniciador es un usuario de IAM, la Cuenta de AWS correspondiente a dicho usuario también tendrá permiso para parar esa carga multiparte. Con las políticas de punto de conexión de VPC, el iniciador de la carga multiparte no obtiene automáticamente permiso para realizar la acción `s3:AbortMultipartUpload` . Además de estas opciones predeterminadas, el propietario del bucket puede permitir a otras entidades principales realizar la acción `s3:AbortMultipartUpload` en un objeto. El propietario del bucket puede denegar la posibilidad de realizar la acción `s3:AbortMultipartUpload` a cualquier entidad principal.
List Parts	Debe tener permiso para realizar la acción `s3:ListMultipartUploadParts` para mostrar las partes de una carga multiparte. De forma predeterminada, el propietario del bucket tiene permiso para mostrar las partes de cualquier carga multiparte en el bucket. El iniciador de la carga multiparte tiene permiso para mostrar las partes de la carga multiparte específica. Si el iniciador de la carga multiparte es un usuario de IAM, la Cuenta de AWS que lo controla tiene permiso para mostrar las partes de esa carga. Además de estas opciones predeterminadas, el propietario del bucket puede permitir a otras entidades principales realizar la acción `s3:ListMultipartUploadParts` en un objeto. El propietario del bucket también puede denegar la posibilidad de realizar la acción `s3:ListMultipartUploadParts` a cualquier entidad principal.

Acción	Permisos necesarios
List Multipart Uploads	Debe tener permiso para realizar la acción `s3:ListBucketMultipartUploads` en un bucket para mostrar las cargas multipartes en curso de ese bucket. Además de esta opción predeterminada, el propietario del bucket puede permitir a otras entidades principales realizar la acción `s3:ListBucketMultipartUploads` en el bucket.
Permisos relacionados con el cifrado y descifrado de AWS KMS	Para realizar una carga multiparte con cifrado mediante una clave de KMS de AWS Key Management Service (AWS KMS), el solicitante debe tener permiso para las acciones `kms:Decrypt` y `kms:GenerateDataKey` en la clave. El solicitante también debe tener permisos para la acción `kms:GenerateDataKey` en la API CreateMultipartUpload. A continuación, el solicitante necesita permisos para realizar la acción `kms:Decrypt` en las API UploadPart y UploadPartCopy. Estos permisos son necesarios, ya que Amazon S3 debe descifrar y leer datos de las partes de archivos cifrados antes de finalizar la carga multiparte. Si su rol o usuario de IAM se encuentran en la misma Cuenta de AWS que la clave de KMS, debe tener estos permisos en la política de claves. Si el usuario o rol de IAM pertenecen a una cuenta distinta de la de la clave de KMS, debe tener los permisos tanto en la política de claves como en el usuario o rol de IAM.

Para obtener información acerca de la relación entre los permisos de la Access Control List (ACL, Lista de control de acceso) y los permisos de las políticas de acceso, consulte Mapeo de permisos de ACL y permisos de política de acceso. Para obtener información acerca de los usuarios, los roles y las prácticas recomendadas de IAM, consulte Identidades (usuarios, grupos de usuarios y roles) de IAM en la Guía del usuario de IAM.

Temas

- Configuración de una política de ciclo de vida del bucket para eliminar cargas multiparte incompletas
- Carga de un objeto con la carga multiparte
- Carga de un directorio utilizando la clase TransferUtility de .NET de alto nivel

- Descripción de cargas multiparte
- Seguimiento de una carga multiparte
- Anulación de la carga multiparte
- Copiar un objeto con la carga multiparte
- Límites de carga multiparte de Amazon S3

Configuración de una política de ciclo de vida del bucket para eliminar cargas multiparte incompletas

Como práctica recomendada, configure una regla de ciclo de vida con la acción `AbortIncompleteMultipartUpload` para reducir los costos de almacenamiento. Para obtener más información sobre cómo anular una carga de varias partes, consulte Anulación de la carga multiparte.

Amazon S3 admite una regla de ciclo de vida del bucket que puede utilizar para que Amazon S3 pare las cargas multiparte que no están completas dentro de un número especificado de días después de iniciarse. Cuando una carga multiparte no se completa en el plazo especificado, pasa a ser apta para una operación de cancelación. Cuando Amazon S3 detiene una carga multiparte y elimina todas las partes asociadas con la carga multiparte. Esta regla se aplica tanto a las cargas multiparte existentes como a las que cree más adelante.

El siguiente es un ejemplo de configuración de ciclo de vida que especifica una regla con la acción `AbortIncompleteMultipartUpload`.

```
<LifecycleConfiguration>
    <Rule>
        <ID>sample-rule</ID>
        <Prefix></Prefix>
        <Status>Enabled</Status>
        <AbortIncompleteMultipartUpload>
          <DaysAfterInitiation>7</DaysAfterInitiation>
        </AbortIncompleteMultipartUpload>
    </Rule>
</LifecycleConfiguration>
```

En el ejemplo, la regla no especifica un valor para el elemento `Prefix` (el prefijo de nombre de clave de objeto). Por lo tanto, la regla se aplica a todos los objetos del bucket para los que se han

iniciado cargas multiparte. Las cargas multiparte que se iniciaron y no se completaron en siete días son aptas para una operación de anulación. La acción de anulación no afecta a las cargas multiparte completadas.

Para obtener más información acerca de la configuración del ciclo de vida del bucket, consulte Administración del ciclo de vida del almacenamiento.

> **Note**
>
> Si la carga multiparte se completa dentro de un periodo especificado de días en la regla, la acción de ciclo de vida `AbortIncompleteMultipartUpload` no aplica (es decir, Amazon S3 no realizará acciones). Además, esta acción no aplica a los objetos. Esta acción de ciclo de vida no elimina objetos. Además, no se te cobrará por la eliminación anticipada de S3 Lifecycle si eliminas cualquier parte de carga multiparte incompleta.

Uso de la consola de S3

Para administrar automáticamente las cargas multiparte incompletas, puede usar la consola de S3 para crear una regla de ciclo de vida para hacer vencer bytes de carga multiparte incompletos del bucket después de un número especificado de días. En el procedimiento siguiente se muestra cómo agregar una regla del ciclo de vida para eliminar cargas multiparte incompletas después de 7 días. Para obtener más información sobre cómo agregar reglas del ciclo de vida, consulte Configuración de un ciclo de vida en un bucket.

Para agregar una regla del ciclo de vida para cancelar las cargas multiparte incompletas que tengan más de 7 días

1. Inicie sesión AWS Management Console Management Console y abra la consola de Amazon S3 en https://console.aws.amazon.com/s3/.
2. En la lista Buckets, seleccione el nombre del bucket para el que desea crear una regla de ciclo de vida.
3. Seleccione la pestaña Management (Administración) y seleccione Create lifecycle rule (Crear regla de ciclo de vida).
4. En Lifecycle rule name (Nombre de regla de ciclo de vida), escriba un nombre para la regla.

   El nombre debe ser único dentro del bucket.
5. Elija el ámbito de la regla de ciclo de vida:

- Para crear una regla del ciclo de vida para todos los objetos con un prefijo específico, elija Limit the scope of this rule using one or more filters (Limitar el alcance de esta regla mediante uno o varios filtros) e ingrese el prefijo en el campo Prefix (Prefijo).
- Para crear una regla de ciclo de vida para todos los objetos del bucket, elija This rule applies to all objects in the bucket (Esta regla se aplica a todos los objetos del bucket) y elija I acknowledge that this rule applies to all objects in the bucket (Reconozco que esta regla se aplica a todos los objetos del bucket).

6. En Lifecycle rule actions (Acciones de reglas del ciclo de vida), seleccione Delete expired object delete markers or incomplete multipart uploads (Eliminar marcadores de eliminación de objetos vencidos o cargas multiparte incompletas).
7. En Delete expired object delete markers or incomplete multipart uploads (Eliminar marcadores de eliminación de objetos vencidos o cargas multiparte incompletas), seleccione Delete incomplete multipart uploads (Eliminar cargas multiparte incompletas).
8. En el campo Number of days (Número de días), ingrese el número de días transcurridos después de eliminar cargas multiparte incompletas (por ejemplo, 7 días).
9. Elija Crear regla.

Uso de la AWS CLI

El siguiente comando `put-bucket-lifecycle-configuration` de la AWS Command Line Interface (AWS CLI) agrega la configuración del ciclo de vida para el bucket especificado. Para usar este comando, sustituya *`user input placeholders`* por la información.

```
aws s3api put-bucket-lifecycle-configuration  \
        --bucket amzn-s3-demo-bucket1  \
        --lifecycle-configuration filename-containing-lifecycle-configuration
```

El siguiente ejemplo muestra cómo agregar una regla del ciclo de vida para anular las cargas multiparte incompletas mediante la AWS CLI. Incluye una configuración del ciclo de vida de JSON de ejemplo para anular las cargas multiparte incompletas que tengan más de 7 días de antigüedad.

Para usar los comandos de la CLI de este ejemplo, sustituya *`user input placeholders`* por la información.

Para agregar una regla del ciclo de vida para cancelar las cargas multiparte incompletas

1. Configure AWS CLI. Para obtener instrucciones, consulte Desarrollo con Amazon S3 mediante la AWS CLI.
2. Guarde la configuración del ciclo de vida de ejemplo siguiente en un archivo (por ejemplo, *lifecycle.json*). La configuración de ejemplo especifica un prefijo vacío y, por lo tanto, se aplica a todos los objetos del bucket. Para restringir la configuración a un subconjunto de objetos, puede especificar un prefijo.

```
{
    "Rules": [
        {
            "ID": "Test Rule",
            "Status": "Enabled",
            "Filter": {
                "Prefix": ""
            },
            "AbortIncompleteMultipartUpload": {
                "DaysAfterInitiation": 7
            }
        }
    ]
}
```

3. Ejecute el siguiente comando de la CLI para establecer esta configuración del ciclo de vida en el bucket.

```
aws s3api put-bucket-lifecycle-configuration   \
--bucket amzn-s3-demo-bucket1  \
--lifecycle-configuration file://lifecycle.json
```

4. Para comprobar que la configuración del ciclo de vida se ha establecido en el bucket, recupere la configuración del ciclo de vida mediante el siguiente comando `get-bucket-lifecycle`.

```
aws s3api get-bucket-lifecycle  \
--bucket amzn-s3-demo-bucket1
```

5. Para eliminar la configuración del ciclo de vida, utilice el comando `delete-bucket-lifecycle` siguiente.

```
aws s3api delete-bucket-lifecycle \
```

```
--bucket amzn-s3-demo-bucket1
```

Carga de un objeto con la carga multiparte

Puede utilizar la carga multiparte para cargar mediante programación un solo objeto en Amazon S3.

Para obtener más información, consulte las siguientes secciones.

Uso de los AWS SDK (API de alto nivel)

Algunos SDK de AWS incluyen una API de alto nivel que simplifica la carga de varias partes al combinar las diferentes operaciones de la API necesarias para completar una carga de varias partes en una sola operación. Para obtener más información, consulte Carga y copia de objetos con la carga multiparte.

Si tiene que detenerse y reanudar cargas de varias partes, variar los tamaños de las partes durante la carga, o si no conoce de antemano el tamaño de los datos, use la API de bajo nivel. Para obtener más información sobre los métodos de la API de bajo nivel para cargas de varias partes que ofrecen funcionalidad adicional, consulte Uso de los AWS SDK (API de bajo nivel).

Java

Para cargar archivos de gran tamaño, use la clase `TransferManager`. Esta operación de la API de alto nivel puede cargar datos de un archivo o una secuencia. También puede configurar opciones avanzadas, como el tamaño de la parte que desea utilizar para la carga multiparte o el número de subprocesos simultáneos que desea utilizar al cargar las partes. También puede establecer propiedades de objeto opcionales, la clase de almacenamiento o la lista de control de acceso (ACL). Puede usar las clases `PutObjectRequest` y `TransferManagerConfiguration` para establecer estas opciones avanzadas.

Cuando sea posible, `TransferManager` intenta usar varios subprocesos para cargar varias partes de una carga a la vez. Esto puede aumentar de forma importante el rendimiento, cuando se utiliza contenido de gran tamaño y un ancho de banda grande.

Además de la funcionalidad de carga de archivos, la clase `TransferManager` permite detener una carga multiparte en curso. Una carga se considera en curso después de que la inicia y hasta que la completa o detiene. `TransferManager` detiene todas las cargas multiparte en curso en un bucket especificado iniciado antes de una fecha y hora especificados.

> **Note**
>
> Cuando se utiliza una secuencia para el origen de datos, la clase `TransferManager` no realiza cargas simultáneas.

En el ejemplo siguiente se carga un objeto con la API Java de carga multiparte de alto nivel (la clase `TransferManager`). Para obtener instrucciones sobre cómo crear y probar una muestra funcional, consulte Introducción en la Guía del desarrollador de AWS SDK for Java.

```
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

Para cargar un archivo a un bucket de S3, utilice la clase `TransferUtility`. Al cargar los datos de un archivo, debe proporcionar el nombre de clave del objeto. Si no lo hace, la API utiliza el nombre del archivo como nombre de clave. Al cargar los datos de una secuencia, debe proporcionar el nombre de clave del objeto.

Para configurar opciones de carga avanzadas, como el tamaño de la parte, el número de subprocesos al cargar las partes simultáneamente, los metadatos, la clase de almacenamiento o la ACL, utilice la clase `TransferUtilityUploadRequest`.

> **Note**
>
> Cuando se utiliza una secuencia para el origen de datos, la clase `TransferUtility` no realiza cargas simultáneas.

En el siguiente ejemplo de código C# se carga un archivo en un bucket de Amazon S3 en varias partes. Muestra cómo utilizar varias sobrecargas `TransferUtility.Upload` para cargar un archivo. Cada llamada sucesiva de carga reemplaza la carga anterior. Para obtener información acerca de cómo configurar y ejecutar ejemplos de código, consulte Introducción al SDK de AWS para .NET en la Guía para desarrolladores del SDK de AWS para .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

Cargue un archivo grande.

```
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

Descargue un archivo grande.

```
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

Cargue un objeto grande mediante un mánager de carga para dividir los datos en partes y cargarlos simultáneamente.

```
// 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

Descargue un objeto grande mediante un mánager de descargas para obtener los datos en partes y descargarlos simultáneamente.

```
// 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

Este tema explica cómo usar la clase `Aws\S3\Model\MultipartUpload\UploadBuilder` de alto nivel de AWS SDK for PHP para cargas de archivos multiparte. Para obtener más información acerca de la API del SDK de AWS para Ruby, consulte SDK de AWS para Ruby, versión 2.

En el siguiente ejemplo de PHP se carga un archivo a un bucket de Amazon S3. El ejemplo muestra cómo configurar parámetros para el objeto `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

En el siguiente ejemplo se carga un objeto con la API Python de carga multiparte de alto nivel (la clase `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
```

Uso de los AWS SDK (API de bajo nivel)

El SDK de AWS expone una API de bajo nivel que se parece bastante a la API de REST de Amazon S3 para cargas multiparte (consulte Carga y copia de objetos con la carga multiparte). Utilice la API de bajo nivel cuando necesite detener y reanudar cargas multiparte, variar los tamaños de las partes durante la carga o si no conoce de antemano el tamaño de los datos de carga. Cuando no tenga estas necesidades, utilice la API de alto nivel (consulte Uso de los AWS SDK (API de alto nivel)).

Java

En el siguiente ejemplo se demuestra cómo usar las clases de Java de bajo nivel para cargar un archivo. Realiza los siguientes pasos:

- Inicia una carga multiparte usando el método `AmazonS3Client.initiateMultipartUpload()` y transmite un objeto `InitiateMultipartUploadRequest`.
- Guarda el ID de carga que devuelve el método `AmazonS3Client.initiateMultipartUpload()`. Facilite este ID de carga para cada operación de carga multiparte subsiguiente.
- Carga las partes del objeto. Para cada parte, llame al método `AmazonS3Client.uploadPart()`. La información de carga de partes debe proporcionarse usando un objeto `UploadPartRequest`.
- Para cada parte, guarda la ETag desde respuesta del método `AmazonS3Client.uploadPart()` en una lista. Utilice los valores de ETag para completar la carga multiparte.
- Llama al método `AmazonS3Client.completeMultipartUpload()` para completar la carga multiparte.

Example

Para obtener instrucciones sobre cómo crear y probar una muestra funcional, consulte Introducción en la Guía del desarrollador de AWS SDK for Java.

```
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

En el siguiente ejemplo de código C# se muestra cómo usar la API de carga multiparte AWS SDK for .NET de bajo nivel para cargar un archivo a un bucket de S3. Para obtener más información sobre cargas multiparte de Amazon S3, consulte Carga y copia de objetos con la carga multiparte.

> **Note**
>
> Cuando usa la API de AWS SDK for .NET para cargar objetos grandes, puede presentarse un tiempo de espera mientras los datos se escriben en la secuencia de la solicitud. Puede establecer un tiempo de espera explícitamente con `UploadPartRequest`.

En el siguiente ejemplo de código C# se carga un archivo a un bucket de S3 con la API de bajo nivel para la carga multiparte. Para obtener información acerca de cómo configurar y ejecutar ejemplos de código, consulte Introducción al SDK de AWS para .NET en la Guía para desarrolladores del SDK de AWS para .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

En este tema, se muestra cómo usar el método `uploadPart` de bajo nivel de la versión 3 de AWS SDK for PHP para cargar un archivo en varias partes. Para obtener más información acerca de la API del SDK de AWS para Ruby, consulte SDK de AWS para Ruby, versión 2.

En el siguiente ejemplo de PHP se carga un archivo a un bucket de Amazon S3 con la API de PHP de bajo nivel para la carga multiparte.

```
 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;
```

Uso de la AWS SDK for Ruby

La versión 3 de AWS SDK for Ruby admite cargas multiparte de Amazon S3 de dos maneras. Para la primera opción, puede utilizar cargas de archivos administrados. Para obtener más información, consulte Uploading Files to Amazon S3 en el Blog para desarrolladores de AWS. Las cargas de archivos administrados son el método recomendado para cargar archivos en un bucket. Proporcionan los siguientes beneficios:

- Administrar cargas multiparte para objetos de más de 15 MB.
- Abra correctamente los archivos en modo binario para evitar problemas de codificación.
- Utilice varios subprocesos para cargar partes de objetos grandes en simultáneo.

Como alternativa, puede utilizar directamente las siguientes operaciones de cliente de carga multiparte:

- create_multipart_upload: inicia una carga multiparte y devuelve un ID de carga.
- upload_part: carga una parte en una carga multiparte.
- upload_part_copy: carga una parte al copiar los datos de un objeto existente como origen de datos.
- complete_multipart_upload: completa una carga multiparte al combinar las partes cargadas previamente.
- abort_multipart_upload: para una carga multiparte.

Uso de la API de REST

En las siguientes secciones de la referencia de la API de Amazon Simple Storage Service, se describe la API de REST para la carga multiparte.

- Initiate Multipart Upload

- Upload Part
- Complete Multipart Upload
- Detener carga multiparte
- List Parts
- List Multipart Uploads

Uso de la AWS CLI

En las siguientes secciones de AWS Command Line Interface (AWS CLI), se describen las operaciones para la carga multiparte.

- Initiate Multipart Upload
- Upload Part
- Upload Part (Copy)
- Complete Multipart Upload
- Abort Multipart Upload
- List Parts
- List Multipart Uploads

También puede utilizar la API de REST para realizar sus propias solicitudes REST o puede utilizar uno de los SDK de AWS. Para obtener más información sobre la API de REST, consulte Uso de la API de REST. Para obtener más información sobre SDKs, consulte Carga de un objeto con la carga multiparte.

Carga de un directorio utilizando la clase TransferUtility de .NET de alto nivel

Puede utilizar la clase `TransferUtility` para cargar todo un directorio. De forma predeterminada, la API solo carga los archivos en la raíz del directorio especificado. Sin embargo, puede especificar que se carguen los archivos de forma recursiva en todos los subdirectorios.

Puede especificar expresiones de filtrado para seleccionar archivos en el directorio especificado según algunos criterios de filtrado. Por ejemplo, para cargar solo archivos .pdf desde un directorio, especifique la expresión de filtrado `"*.pdf"`.

Cuando carga archivos desde un directorio, no puede especificar los nombres de clave de los objetos resultantes. Amazon S3 construye los nombres de clave usando la ruta del archivo original. Por ejemplo, suponga que tiene un directorio llamado `c:\myfolder` con la siguiente estructura:

Example

```
C:\myfolder
      \a.txt
      \b.pdf
      \media\
             An.mp3
```

Cuando carga este directorio, Amazon S3 utiliza los siguientes nombres de clave:

Example

```
a.txt
b.pdf
media/An.mp3
```

Example

En el siguiente ejemplo de código C# se carga un directorio a un bucket de Amazon S3. Muestra cómo utilizar varias sobrecargas `TransferUtility.UploadDirectory` para cargar el directorio. Cada llamada sucesiva de carga reemplaza la carga anterior. Para obtener información acerca de cómo configurar y ejecutar ejemplos de código, consulte Introducción al SDK de AWS para .NET en la Guía para desarrolladores del SDK de AWS para .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);
            }
        }
    }
}
```

Descripción de cargas multiparte

Puede utilizar los AWS SDK (API de bajo nivel) para recuperar una lista de cargas multiparte en curso en Amazon S3.

Descripción de cargas multiparte con el AWS SDK (API de bajo nivel)

Java

Las siguientes tareas lo guían a través del proceso de uso de las clases de Java de nivel bajo para enumerar las cargas multipartes en proceso en un bucket.

Proceso de descripción de cargas multiparte de API de bajo nivel

1	Cree una instancia de la clase `ListMultipartUploadsRequest` y proporcione el nombre del bucket.
2	Ejecute el método `AmazonS3Client.listMultipartUploads`. El método devuelve una instancia de la clase `MultipartUploadListing` que le brinda información sobre las cargas multipartes en proceso.

En el siguiente ejemplo de código Java se muestran las tareas anteriores.

Example

```
ListMultipartUploadsRequest allMultpartUploadsRequest =
```

```
     new ListMultipartUploadsRequest(existingBucketName);
MultipartUploadListing multipartUploadListing =
     s3Client.listMultipartUploads(allMultpartUploadsRequest);
```

.NET

Para enumerar todas las cargas multiparte en curso en un bucket concreto, use la clase `ListMultipartUploadsRequest` de la API de carga multiparte de bajo nivel de AWS SDK for .NET. El método `AmazonS3Client.ListMultipartUploads` devuelve una instancia de la clase `ListMultipartUploadsResponse` que ofrece información sobre las cargas multiparte en curso.

Una carga multiparte en curso es una carga multiparte que se ha iniciado mediante una solicitud de carga multiparte pero que aún no se ha completado o detenido. Para obtener más información acerca de las cargas multiparte en Amazon S3, consulte Carga y copia de objetos con la carga multiparte.

En el siguiente ejemplo de código C# se muestra cómo usar AWS SDK for .NET para enumerar todas las cargas multiparte en curso en un bucket. Para obtener información acerca de cómo configurar y ejecutar ejemplos de código, consulte Introducción al SDK de AWS para .NET en la Guía para desarrolladores del SDK de AWS para .NET.

```
ListMultipartUploadsRequest request = new ListMultipartUploadsRequest
{
  BucketName = bucketName // Bucket receiving the uploads.
};

ListMultipartUploadsResponse response = await
 AmazonS3Client.ListMultipartUploadsAsync(request);
```

PHP

Este tema muestra cómo usar las clases de API de bajo nivel de la versión 3 de AWS SDK for PHP para crear una lista de todas las cargas multiparte en curso en un bucket. Para obtener más información acerca de la API del SDK de AWS para Ruby, consulte SDK de AWS para Ruby, versión 2.

El siguiente ejemplo de PHP demuestra cómo enumerar todas las cargas multiparte en curso en un bucket.

```
 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());
```

Descripción de cargas multiparte con la API de REST

Las siguientes secciones de la Referencia de API de Amazon Simple Storage Service especifican la API de REST para describir cargas multiparte:

- ListParts: describe las partes cargadas para una carga multiparte específica.
- ListMultipartUploads: describe las cargas multiparte en curso.

Descripción de cargas multiparte con la AWS CLI

En las siguientes secciones de AWS Command Line Interface, se describen las operaciones para visualizar listas de las cargas multiparte.

- list-parts: describe las partes cargadas para una carga multiparte específica.
- list-multipart-uploads: describe las cargas multiparte en curso.

Seguimiento de una carga multiparte

La API de carga multiparte de alto nivel proporciona una interfaz de escucha `ProgressListener`, para realizar un seguimiento del progreso de carga cuando se carga un objeto en Amazon S3. Los

eventos de progreso se producen de forma periódica y notifican la transferencia de bytes al agente de escucha.

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

El siguiente código Java carga un archivo y utiliza la interfaz `ProgressListener` para hacer el seguimiento del progreso de la carga. Para obtener instrucciones sobre cómo crear y probar una muestra funcional, consulte Introducción en la Guía del desarrollador de AWS SDK for Java.

```
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

En el siguiente ejemplo de código C# se carga un archivo a un bucket de S3 mediante la clase `TransferUtility` y se sigue el progreso de la carga. Para obtener información acerca de cómo configurar y ejecutar ejemplos de código, consulte Introducción al SDK de AWS para .NET en la Guía para desarrolladores del SDK de AWS para .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);
        }
    }
}
```

Anulación de la carga multiparte

Comience a cargar las partes luego de iniciar una carga multiparte. Amazon S3 almacena estas partes, pero crea el objeto de las partes solo después de que usted las carga y envía una solicitud `successful` para completar la carga multiparte (debe comprobar si la solicitud para completar una carga multiparte es correcta). Al recibir la solicitud de carga multiparte completa, Amazon S3 combina las partes y crea un objeto. Si no envía correctamente la solicitud de carga multiparte completa, Amazon S3 no combina las partes y no crea un objeto.

Se facturará todo el almacenamiento asociado con las partes cargadas. Para obtener más información, consulte Carga multiparte y precios. Por lo tanto, es importante que complete la carga multiparte para que se cree el objeto o detenga la carga multiparte a fin de eliminar las partes cargadas.

Puede detener una carga multiparte en curso en Amazon S3 mediante AWS Command Line Interface (AWS CLI), la API de REST o los SDK de AWS. También puede detener una carga multiparte incompleta mediante una configuración de ciclo de vida del bucket.

Uso de los AWS SDK (API de alto nivel)

Java

La clase `TransferManager` proporciona el método `abortMultipartUploads` para detener cargas multiparte en curso. Una carga se considera en curso después de que la inicia y hasta que la completa o detiene. Cuando provee un valor de `Date`, esta API detiene todas las cargas multiparte, en ese bucket, que se iniciaron antes de la `Date` especificada y que aún están en curso.

Las siguientes tareas lo guían a través del proceso de uso de las clases de Java de alto nivel para detener cargas multiparte.

Proceso de detención de cargas multiparte de la API de alto nivel

1	Cree una instancia de la clase `TransferManager` .
2	Ejecute el método `TransferManager.abortMultipartUploads` y pase el nombre del bucket y un valor de `Date`.

El siguiente código Java detiene todas las cargas multiparte en curso que se iniciaron en un bucket específico hace más de una semana. Para obtener instrucciones sobre cómo crear y probar una muestra funcional, consulte Introducción en la Guía del desarrollador de AWS SDK for Java.

```
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**
>
> También puede detener una carga multiparte específica. Para obtener más información, consulte Uso de los AWS SDK (API de bajo nivel).

.NET

En el siguiente ejemplo de código C# se detienen todas las cargas multiparte en curso que se iniciaron en un bucket específico hace una semana. Para obtener información acerca de cómo configurar y ejecutar ejemplos de código, consulte Introducción al SDK de AWS para .NET en la Guía para desarrolladores del SDK de AWS para .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**
>
> También puede detener una carga multiparte específica. Para obtener más información, consulte Uso de los AWS SDK (API de bajo nivel).

Uso de los AWS SDK (API de bajo nivel)

Puede detener una carga multiparte en curso si llama al método `AmazonS3.abortMultipartUpload`. Este método elimina cualquier parte cargada Amazon S3 y libera los recursos. Deberá proporcionar el ID de carga, el nombre del bucket y el nombre de clave. En el siguiente ejemplo de código Java se muestra cómo detener una carga multiparte en curso.

Para detener una carga multiparte, debe proporcionar el ID de carga y los nombres de clave y bucket que se usaron en la carga. Luego de haber detenido una carga multiparte, no puede usar el ID de carga para cargar partes adicionales. Para obtener más información acerca de las cargas multiparte en Amazon S3, consulte Carga y copia de objetos con la carga multiparte.

Java

En el siguiente ejemplo de código Java se detiene una carga multiparte en curso.

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**
>
> En lugar de una carga multiparte específica, puede detener todas las cargas multiparte iniciadas antes de un periodo de tiempo específico que siguen en proceso. Esta operación de limpieza es útil para detener las cargas multiparte antiguas que inició pero que no se completaron o detuvieron. Para obtener más información, consulte Uso de los AWS SDK (API de alto nivel).

.NET

En el siguiente ejemplo en C# se muestra cómo detener una carga multiparte. Para ver una muestra completa de código C# que incluye el código siguiente, consulte Uso de los AWS SDK (API de bajo nivel).

```
AbortMultipartUploadRequest abortMPURequest = new AbortMultipartUploadRequest
{
    BucketName = existingBucketName,
    Key = keyName,
    UploadId = initResponse.UploadId
};
await AmazonS3Client.AbortMultipartUploadAsync(abortMPURequest);
```

También puede anular todas las cargas multiparte en curso que se iniciaron antes de un periodo de tiempo específico. Esta operación de limpieza resulta útil para anular cargas multiparte que no se finalizaron o que se anularon. Para obtener más información, consulte Uso de los AWS SDK (API de alto nivel).

PHP

En este ejemplo, se muestra cómo utilizar una clase de la versión 3 de AWS SDK for PHP a fin de anular una carga multiparte que está en curso. Para obtener más información acerca de la API del SDK de AWS para Ruby, consulte SDK de AWS para Ruby, versión 2. El ejemplo es el método `abortMultipartUpload()`.

Para obtener más información acerca de la API del SDK de AWS para Ruby, consulte SDK de AWS para Ruby, versión 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,
]);
```

Uso de la API de REST

Para obtener más información sobre el uso de la API de REST a fin de detener una carga multiparte, consulte AbortMultipartUpload en la Referencia de API de Amazon Simple Storage Service.

Uso de la AWS CLI

Para obtener más información acerca del uso de la AWS CLI a fin de detener una carga multiparte, consulte abort-multipart-upload en la Referencia de comandos de la AWS CLI.

Copiar un objeto con la carga multiparte

En esta sección se muestran ejemplos sobre cómo copiar objetos superiores a 5 GB con la Application Programming Interface (API, Interfaz de programación de aplicaciones) de carga multiparte. Puede copiar objetos inferiores a 5 GB con una sola operación. Para obtener más información, consulte Copia, traslado y cambio de nombre de objetos.

Uso de los AWS SDK

Para copiar un objeto con la API de bajo nivel, siga estos pasos:

- Inicie una carga multiparte con el llamado al método `AmazonS3Client.initiateMultipartUpload()`.
- Guarde el ID de carga del objeto de respuesta que devuelve el método `AmazonS3Client.initiateMultipartUpload()`. Facilite este ID de carga para cada operación de carga de parte.
- Copie todas las partes. Para cada parte que necesite copiar, cree una nueva instancia de la clase `CopyPartRequest`. Proporcione la información de parte, incluidos los nombres de los bucket de origen y destino, las claves de los objetos de origen y de destino, los ID de carga, las ubicaciones del primer y último byte de la parte y el número de parte.
- Guarde las respuestas de las llamadas del método `AmazonS3Client.copyPart()`. Cada respuesta incluye el valor de `ETag` y el número de parte de la parte cargada. Necesitará esta información para completar la carga multiparte:
- Llame al método `AmazonS3Client.completeMultipartUpload()` para completar la operación de copia.

Java

Example

En el siguiente ejemplo se muestra cómo usar la API de Java de bajo nivel de Amazon S3 para realizar una copia multiparte. Para obtener instrucciones sobre cómo crear y probar una muestra funcional, consulte Introducción en la Guía del desarrollador de AWS SDK for Java.

```
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

En el siguiente ejemplo de código C#, se muestra cómo usar AWS SDK for .NET para copiar un objeto de Amazon S3 mayor que 5 GB de una ubicación de origen a otra, como, por ejemplo, de un bucket a otro. Para copiar objetos menores de 5 GB, use el procedimiento de copia de una sola operación descrito en Uso de los AWS SDK. Para obtener más información acerca de las cargas multiparte en Amazon S3, consulte Carga y copia de objetos con la carga multiparte.

En este ejemplo, se muestra cómo copiar un objeto de Amazon S3 con un tamaño superior a 5 GB de un bucket de S3 a otro con la API de carga multiparte de AWS SDK for .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 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);
            }
        }
    }
}
```

Uso de la API de REST

Las siguientes secciones de la referencia de API de Amazon Simple Storage Service describen la API de REST para la carga multiparte. Para copiar un objeto existente, utilice la API Upload Part (Copy) y agregue el encabezado de solicitud `x-amz-copy-source` a su solicitud a fin de especificar el objeto de origen.

- Initiate Multipart Upload
- Upload Part
- Upload Part (Copy)
- Complete Multipart Upload
- Abort Multipart Upload
- List Parts
- List Multipart Uploads

Puede utilizar estas API para realizar sus propias solicitudes REST o puede utilizar uno de los SDK que ofrecemos. Para obtener más información sobre el uso de la carga multiparte con la AWS CLI,

consulte Uso de la AWS CLI. Para obtener más información sobre SDKs, consulte Compatibilidad de AWS SDK con cargas multiparte.

Límites de carga multiparte de Amazon S3

En la siguiente tabla se proporcionan las especificaciones principales de la carga multiparte. Para obtener más información, consulte Carga y copia de objetos con la carga multiparte.

Elemento	Especificación
Tamaño máximo de objeto	5 TiB
Cantidad máxima de partes por cada carga	10 000
Números de parte	De 1 a 10 000 (inclusive)
Tamaño de parte	De 5 MiB a 5 GiB. No hay límite de tamaño mínimo en la última parte de la carga multiparte.
Cantidad máxima de partes devueltas para una solicitud de lista de partes	1 000
Cantidad máxima de cargas multipartes devueltas en una solicitud de lista de cargas multipartes	1 000

Copia, traslado y cambio de nombre de objetos

La operación `CopyObject` crea una copia de un objeto que ya está almacenado en Amazon S3.

Puede crear una copia de un objeto de hasta 5 GB en una única operación atómica. Sin embargo, para copiar un objeto mayor de 5 GB, debe usar una carga multiparte. Para obtener más información, consulte the section called “Copiar un objeto”.

Con la operación `CopyObject`, puede:

- Cree copias adicionales de objetos.
- Cambie el nombre de objetos copiándolos y eliminando los originales
- Copie o mueva objetos de un bucket a otro, incluso entre Regiones de AWS (por ejemplo, de `us-west-1` a `eu-west-2`). Cuando mueve un objeto, Amazon S3 copia el objeto en el destino especificado y, a continuación, elimina el objeto original.

> **Note**
>
> La copia o el tralsado objetos entre Regiones de AWS genera cargos por uso de ancho de banda. Para obtener más información, consulte Precios de Amazon S3.

- Cambie metadatos de objetos. Cada objeto de Amazon S3 tiene metadatos. Estos metadatos son un conjunto de pares nombre-valor. Puede establecer los metadatos de un objeto en el momento en el que lo sube. Tras cargar el objeto, no puede modificar sus metadatos. La única manera de modificar los metadatos de un objeto es realizar una copia del mismo y configurar sus metadatos. Para ello, en la operación de copia, ajuste el mismo objeto en el origen y el destino.

  Algunos los metadatos del objeto son metadatos del sistema y otros definidos por el usuario. Puede controlar algunos de los metadatos del sistema. Por ejemplo, puede controlar la clase de almacenamiento y el tipo de cifrado del servidor que se utilizará para el objeto. Cuando copia un objeto, los metadatos del sistema controlados por el usuario y los metadatos definidos por el usuario también se copian. Amazon S3 reinicia los metadatos controlados por el sistema. Por ejemplo, cuando se copia un objeto, Amazon S3 restablece la fecha de creación del objeto copiado. No necesita establecer ninguno de estos valores de metadatos controlados por el sistema en su solicitud de copia.

  Al copiar un objeto, puede decidir actualizar algunos de los valores de los metadatos. Por ejemplo, si su objeto de origen está configurado para usar almacenamiento S3 estándar, podría elegir usar S3 Intelligent-Tiering para la copia del objeto. También podría decidir alterar algunos de los valores de metadatos definidos por el usuario presentes en el objeto de origen. Si decide actualizar cualquiera de los metadatos del objeto configurables por el usuario (definidos por el sistema o por el usuario) durante la copia, debe especificar explícitamente todos los metadatos configurables por el usuario presentes en el objeto de origen de la solicitud, incluso aunque solo cambie uno de los valores de metadatos.

  Para obtener más información acerca de los metadatos del objeto, consulte Trabajar con metadatos de objeto.

Copia de objetos archivados y restaurados

Si el objeto de origen está archivado en S3 Glacier Flexible Retrieval o S3 Glacier Deep Archive, deberá restaurar primero una copia temporal antes de poder copiar el objeto a otro bucket. Para obtener información acerca del archivado de objetos, consulte Transición a las clases de almacenamiento S3 Glacier Flexible Retrieval y S3 Glacier Deep Archive (archivo de objetos).

La operación Copy (Copiar) no se admite en la consola de Amazon S3 para los objetos restaurados de las clases de almacenamiento S3 Glacier Flexible Retrieval o S3 Glacier Deep Archive. Para copiar estos objetos restaurados, utilice la AWS Command Line Interface (AWS CLI), los SDK de AWS o la API de REST de Amazon S3.

Copia de objetos cifrados

Amazon S3 cifra automáticamente todos los objetos nuevos que se copian a un bucket de S3. Si no especifica la información de cifrado en su solicitud de copia, la configuración de cifrado del objeto de destino se establece con la configuración de cifrado predeterminada del bucket de destino. De forma predeterminada, todos los buckets tienen un nivel de configuración de cifrado básico que utiliza el cifrado del lado del servidor con claves administradas de Amazon S3 (SSE-S3). Si el bucket de destino tiene una configuración de cifrado predeterminada que utiliza el cifrado del lado del servidor con una clave de AWS Key Management Service (AWS KMS) (SSE-KMS) o una clave de cifrado proporcionada por el cliente (SSE-C), Amazon S3 utiliza la clave de KMS correspondiente o una clave proporcionada por el cliente para cifrar la copia del objeto de destino.

Al copiar un objeto, si desea utilizar un tipo de configuración de cifrado diferente para el objeto de destino, puede solicitar que Amazon S3 cifre el objeto de destino con una clave de KMS, una clave administrada de Amazon S3 o una clave proporcionada por el cliente. Si la configuración de cifrado de su solicitud es diferente de la configuración de cifrado predeterminada del bucket de destino, tendrá prioridad la configuración de cifrado de su solicitud. Si el objeto de origen de la copia se ha cifrado con SSE-C, debe facilitar la información de cifrado necesaria en la solicitud, de modo que Amazon S3 pueda descifrar el objeto para copiarlo. Para obtener más información, consulte Protección de los datos mediante el cifrado.

Uso de sumas de comprobación al copiar objetos

Al copiar objetos, puede optar por utilizar un algoritmo de suma de comprobación diferente para el objeto. Tanto si elige utilizar el mismo algoritmo como uno nuevo, Amazon S3 calcula un nuevo valor de suma de comprobación después de copiar el objeto. Amazon S3 no copia directamente el valor de la suma de comprobación. El valor de la suma de comprobación de los objetos que se han

cargado mediante cargas multiparte podría cambiar. Para obtener más información sobre el cálculo de la suma de comprobación, consulte Uso de sumas de comprobación a nivel de parte para cargas multiparte.

Copia de varios objetos en una sola solicitud

Para copiar más de un objeto de Amazon S3 con una sola solicitud, puede utilizar las operaciones por lotes de Amazon S3. Proporcione a Operaciones por lotes de S3 una lista de objetos en los que operar. Operaciones por lotes de S3 llama a la operación de la API respectiva para realizar la operación especificada. Un solo trabajo de operaciones por lotes puede realizar la operación especificada en miles de millones de objetos con exabytes de datos.

Operaciones por lotes de S3 realiza un seguimiento del avance, envía notificaciones y guarda un informe de finalización de todas las acciones, por lo que proporciona una experiencia sin servidor, auditable, completamente administrada. Puede emplear Operaciones por lotes de S3 a través de la consola de Amazon S3, AWS CLI, los SDK de AWS o la API de REST. Para obtener más información, consulte the section called “Conceptos básicos de las operaciones por lotes”.

Copia de objetos a buckets de directorio

Para obtener información sobre cómo copiar un objeto a un bucket de directorio, consulte Copiar un objeto en un bucket de directorio. Para obtener información sobre el uso de la clase de almacenamiento Amazon S3 Express One Zone con buckets de directorio, consulte ¿Qué es S3 Express One Zone? y Buckets de directorio.

Para copiar un objeto

Para copiar un objeto, utilice los métodos siguientes.

Uso de la consola de S3

> **Note**
>
> - Cuando copia un objeto utilizando la consola de Amazon S3, debe contar con el permiso `s3:ListAllMyBuckets`. La consola necesita este permiso para validar la operación de Copy (Copia). Para ver políticas de ejemplo que conceden estos permisos, consulte the section called “Ejemplos de políticas basadas en identidades”.
>
>   Si va a copiar un objeto que tiene etiquetas definidas por el usuario, también debe tener el permiso `s3:GetObjectTagging`. Si va a copiar un objeto que no tiene etiquetas

definidas por el usuario pero que tiene un tamaño superior a 16 MB, también debe tener el permiso S3:`GetObjectTagging`.

Si la política de bucket de destino deniega la acción `s3:GetObjectTagging`, el objeto se copiará sin las etiquetas definidas por el usuario y aparecerá un error.

- Los objetos cifrados con claves de cifrado proporcionadas por el cliente (SSE-C) no se pueden copiar usando la consola de S3. Para copiar objetos cifrados con SSE-C, utilice la AWS CLI, el SDK de AWS o la API REST de Amazon S3.
- La consola de Amazon S3 no admite la copia entre regiones de objetos cifrados con SSE-KMS. Para copiar objetos cifrados con SSE-KMS entre regiones, utilice la AWS CLI, el SDK de AWS o la API de REST de Amazon S3.

Para copiar un objeto

1. Inicie sesión AWS Management Console Management Console y abra la consola de Amazon S3 en https://console.aws.amazon.com/s3/.
2. En el panel de navegación de la izquierda, seleccione Buckets y, a continuación, la pestaña General purpose buckets (Buckets de uso general). Desplácese hasta el bucket o la carpeta de Amazon S3 que contiene los objetos que desea copiar.
3. Seleccione la casilla de verificación situada a la izquierda de los nombres de los objetos que desea copiar.
4. En el menú Actions (Acciones), elija Copy (Copiar) en la lista de opciones que aparece.
5. Seleccione el tipo de destino y la cuenta de destino. Para especificar la ruta de destino, seleccione Browse S3 (Examinar S3), desplácese hasta el destino y active la casilla de verificación situada a la izquierda del destino. Seleccione Elegir destino en la esquina inferior derecha.

   También puede escribir la ruta de destino.
6. Si no tiene activado el control de versiones del bucket, es posible que se le pida que confirme que los objetos que tengan el mismo nombre se deben sobrescribir. Si así es, seleccione la casilla de verificación y continúe. Si quiere mantener todas las versiones de los objetos en este bucket, seleccione Enable Bucket Versioning (Habilitar control de versiones de bucket). También puede actualizar las propiedades predeterminadas de cifrado y de bloqueo de objetos de S3.
7. En Additional checksums (Sumas de comprobación adicionales), elija si desea copiar los objetos utilizando la función de suma de comprobación existente o sustituir la función de suma de

comprobación existente por otra nueva. Al cargar los objetos, tenía la opción de especificar el algoritmo de suma de comprobación que se utilizó para verificar la integridad de los datos. Al copiar el objeto, tiene la opción de elegir una nueva función. Si no especificó originalmente una suma de comprobación adicional, puede utilizar esta sección de las opciones de copia para agregar una.

> **Note**
>
> Incluso si opta por utilizar la misma función de suma de comprobación, el valor de la suma de comprobación podría cambiar si copia el objeto y este tiene un tamaño superior a 16 MB. El valor de la suma de comprobación podría cambiar debido a cómo se calculan las sumas de comprobación para las cargas multiparte. Para obtener más información acerca de cómo puede cambiar la suma de comprobación al copiar el objeto, consulte Uso de sumas de comprobación a nivel de parte para cargas multiparte.

Para cambiar la función de suma de comprobación, elija Reemplazar por una nueva función de suma de comprobación. Seleccione la nueva función de suma de comprobación en el recuadro. Cuando se copia el objeto, la nueva suma de comprobación se calcula y almacena con el algoritmo especificado.

8. En la esquina inferior derecha, elija Copiar. Amazon S3 copia el objeto en el destino.

Uso de los AWS SDK

Los ejemplos de esta sección demuestran cómo copiar objetos de hasta 5 GB en una única operación. Para copiar objetos mayores de 5 GB, debe usar una carga multiparte. Para obtener más información, consulte Copiar un objeto con la carga multiparte.

Java

Example

En el siguiente ejemplo, se copia un objeto en Amazon S3 mediante AWS SDK for Java. Para obtener instrucciones sobre cómo crear y probar una muestra funcional, consulte Introducción en la Guía del desarrollador de AWS SDK for Java.

```
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

En el siguiente ejemplo de código C#, se utiliza AWS SDK for .NET de alto nivel para copiar objetos con un tamaño que puede alcanzar los 5 GB en una sola operación. Para objetos que

sean mayores de 5 GB, use el ejemplo de copia de carga multiparte que se describe en Copiar un objeto con la carga multiparte.

En este ejemplo se realiza una copia de un objeto que tiene un máximo de 5 GB. Para obtener información acerca de cómo configurar y ejecutar ejemplos de código, consulte Introducción al SDK de AWS para .NET en la Guía para desarrolladores del SDK de AWS para .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

En este tema, se detallará el proceso de uso las clases de la versión 3 de AWS SDK for PHP que permiten copiar un único objeto o varios de ellos dentro de Amazon S3, de un bucket a otro o dentro del mismo bucket.

Para obtener más información acerca de la API del SDK de AWS para Ruby, consulte SDK de AWS para Ruby, versión 2.

El siguiente ejemplo de código PHP ilustra el uso del método `copyObject()` para copiar un único objeto en Amazon S3. También se muestra cómo realizar varias copias de un objeto mediante un lote de llamadas a `CopyObject` con el método `getcommand()`.

Copia de objetos

1	Cree una instancia de un cliente de Amazon S3 con el constructor de clase `Aws \S3\S3Client` .
2	Para realizar varias copias de un objeto, ejecute un lote de llamadas al método del cliente de Amazon S3 getCommand(), heredado de la clase Aws\CommandInterfa

	ce. Debe proporcionar el comando `CopyObject` como primer argumento y una matriz que contenga el bucket de origen, el nombre de la clave de origen, el bucket de destino y el nombre de la clave de destino como segundo argumento.

```
 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

Las siguientes tareas lo guiarán a través del uso de las clases de Ruby para copiar un objeto en Amazon S3 de un bucket a otro o dentro del mismo bucket.

Copia de objetos

1	Utilice la gema modularizada de Amazon S3 para la versión 3 de AWS SDK for Ruby, solicite `aws-sdk-s3` y proporcione sus credenciales de AWS. Para obtener más información acerca de cómo proporcionar sus credenciales, consulte Realización de solicitudes con las credenciales de usuario de IAM o Cuenta de AWS.
2	Brinde la información de solicitud, como el nombre del bucket de origen, el nombre de la clave de origen, el nombre de bucket de destino y la clave de destino.

En el siguiente ejemplo de código Ruby se muestran las tareas anteriores utilizando el método `#copy_object` para copiar un objeto de un bucket a otro.

```
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__
```

Uso de la API de REST

En este ejemplo se describe cómo copiar un objeto con la API de REST de Amazon S3. Para obtener más información sobre la API de REST, consulte CopyObject.

En este ejemplo se copia el objeto `flotsam` desde el bucket *amzn-s3-demo-bucket1* al objeto `jetsam` del bucket *amzn-s3-demo-bucket2*, conservando sus metadatos.

```
PUT /jetsam HTTP/1.1
Host: amzn-s3-demo-bucket2.s3.amazonaws.com
x-amz-copy-source: /amzn-s3-demo-bucket1/flotsam
Authorization: AWS AKIAIOSFODNN7EXAMPLE:ENoSbxYByFA0UGLZUqJN5EUnLDg=
Date: Wed, 20 Feb 2008 22:12:21 +0000
```

La firma se generó a partir de la siguiente información.

```
PUT\r\n
\r\n
\r\n
Wed, 20 Feb 2008 22:12:21 +0000\r\n

x-amz-copy-source:/amzn-s3-demo-bucket1/flotsam\r\n
/amzn-s3-demo-bucket2/jetsam
```

Amazon S3 devuelve la siguiente respuesta, que especifica el ETag del objeto y cuándo se modificó por última vez.

```
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>
```

Uso de la AWS CLI

También puede usar la AWS Command Line Interface (AWS CLI) para copiar un objeto de S3. Para obtener más información, consulte copy-object en la Referencia de los comandos de AWS CLI.

Para obtener información sobre AWS CLI, consulte ¿Qué es AWS Command Line Interface? en la Guía del usuario de AWS Command Line Interface.

Para mover un objeto

Para mover un objeto, utilice los métodos siguientes.

Uso de la consola de S3

> **Note**
>
> - Si va a trasladar un objeto que tiene etiquetas definidas por el usuario, también debe tener el permiso `s3:GetObjectTagging`. Si va a trasladar un objeto que no tiene etiquetas definidas por el usuario pero que tiene un tamaño superior a 16 MB, también debe tener el permiso S3:`GetObjectTagging`.
>
> Si la política de bucket de destino deniega la acción `s3:GetObjectTagging`, el objeto se trasladará sin las etiquetas definidas por el usuario y aparecerá un error.
> - Los objetos cifrados con claves de cifrado proporcionadas por el cliente (SSE-C) no se pueden mover con la consola Amazon S3. Para mover objetos cifrados con SSE-C, utilice la AWS CLI, los SDK de AWS o la API de REST de Amazon S3.
> - Al mover carpetas, espere a que finalice la operación Move (Mover) antes de realizar cambios adicionales en las carpetas.
> - No puede usar los alias de los puntos de acceso S3 como origen o destino para las operaciones de Move (Mover) en la consola de Amazon S3.

Para mover un objeto

1. Inicie sesión AWS Management Console Management Console y abra la consola de Amazon S3 en https://console.aws.amazon.com/s3/.

2. En el panel de navegación de la izquierda, seleccione Buckets y, a continuación, la pestaña General purpose buckets (Buckets de uso general). Desplácese hasta el bucket o carpeta de Amazon S3 que contiene los objetos que quiera mover.
3. Seleccione la casilla situada a la izquierda del nombre de cada uno de los objetos que quiera mover.
4. En el menú Actions (Acciones), elija Move (Mover).
5. Para especificar la ruta de destino, seleccione Examinar S3, desplácese hasta el destino y active la casilla de verificación situada a la izquierda del destino. Seleccione Elegir destino en la esquina inferior derecha.

   También puede escribir la ruta de destino.
6. Si no tiene activado el control de versiones del bucket, es posible que se le pida que confirme que los objetos que tengan el mismo nombre se deben sobrescribir. Si así es, seleccione la casilla de verificación y continúe. Si quiere mantener todas las versiones de los objetos en este bucket, seleccione Enable Bucket Versioning (Habilitar control de versiones de bucket). También puede actualizar las propiedades predeterminadas de cifrado y de bloqueo de objetos.
7. En la esquina inferior derecha, elija Move (Mover). Amazon S3 mueve los objetos al destino.

> **Note**
>
> - Esta acción crea una copia de todos los objetos especificados con parámetros actualizados, actualiza la fecha de última modificación en la ubicación especificada y agrega un marcador de eliminación al objeto original.
> - Esta acción actualiza los metadatos para el control de versiones de bucket, el cifrado, las características de bloqueo de objetos y los objetos archivados.

Uso de la AWS CLI

También puede usar AWS Command Line Interface (AWS CLI) para copiar un objeto de S3. Para obtener más información, consulte mv en la Referencia de los comandos de AWS CLI.

Para obtener información sobre AWS CLI, consulte ¿Qué es AWS Command Line Interface? en la Guía del usuario de AWS Command Line Interface.

Para cambiar el nombre de un objeto

Utilice el siguiente procedimiento para cambiar el nombre de un objeto.

> **Note**
>
> - Al cambiar el nombre de un objeto se crea una copia del objeto con una nueva fecha de última modificación y, a continuación, se agrega un marcador de eliminación al objeto original.
> - Los ajustes del bucket para el cifrado predeterminado se aplican automáticamente a cualquier objeto especificado que no esté cifrado.
> - No puede usar la consola de Amazon S3 para cambiar el nombre de los objetos con claves de cifrado (SSE-C) proporcionadas por el cliente. Para cambiar el nombre de los objetos cifrados con SSE-C, utilice la AWS CLI, los AWS SDK o la API de REST de Amazon S3 para copiar esos objetos con nombres nuevos.
> - Si este bucket utiliza el ajuste Aplicada al propietario del bucket para Propiedad de objetos de S3, las listas de control de acceso (ACL) a los objetos no se copiarán.
> - Si va a cambiar el nombre de un objeto que tiene etiquetas definidas por el usuario, también debe tener el permiso `s3:GetObjectTagging`. Si va a cambiar el nombre de un objeto que no tiene etiquetas definidas por el usuario pero que tiene un tamaño superior a 16 MB, también debe tener el permiso S3:`GetObjectTagging`.
>
>   Si la política de bucket de destino deniega la acción `s3:GetObjectTagging`, se cambiará el nombre del objeto, pero las etiquetas definidas por el usuario se eliminarán del objeto y aparecerá un error.

Para cambiar el nombre de un objeto

1. Inicie sesión AWS Management Console Management Console y abra la consola de Amazon S3 en https://console.aws.amazon.com/s3/.
2. En el panel de navegación de la izquierda, seleccione Buckets y, a continuación, la pestaña General purpose buckets (Buckets de uso general). Desplácese hasta el bucket o la carpeta de Amazon S3 que contiene el objeto cuyo nombre quiere cambiar.
3. Seleccione la casilla de verificación situada a la izquierda del nombre del objeto que quiere cambiar.

4. En el menú Acciones, seleccione Cambiar el nombre.
5. En el cuadro Nombre del nuevo objeto, introduzca el nuevo nombre del objeto.
6. En la esquina inferior derecha, elija Guardar cambios. Amazon S3 cambia el nombre de su objeto.

Descarga de objetos

En esta sección se explica cómo descargar objetos desde un bucket de Amazon S3. Con Amazon S3, puede almacenar objetos en uno o varios buckets, y cada objeto individual puede tener un tamaño de hasta 5 TB. Se puede acceder en tiempo real a cualquier objeto de Amazon S3 que no esté archivado. Los objetos archivados, sin embargo, deben restaurarse antes de poder descargarse. Para obtener más información sobre cómo descargar objetos archivados, consulte the section called “Descarga de registros archivados”.

Puede descargar un solo objeto utilizando la consola de Amazon S3, AWS Command Line Interface (AWS CLI), los SDK de AWS o la API de REST de Amazon S3. Para descargar un objeto de S3 sin escribir código ni ejecutar ningún comando, utilice la consola de S3. Para obtener más información, consulte the section called “Descarga de un objeto”.

Para descargar varios objetos, utilice AWS CloudShell, la AWS CLI o los SDK de AWS. Para obtener más información, consulte the section called “Descarga de varios objetos”.

Si necesita descargar parte de un objeto, utilice parámetros adicionales con la AWS CLI o la API de REST para especificar únicamente los bytes que desee descargar. Para obtener más información, consulte the section called “Para descargar parte de un objeto”.

Si necesita descargar un objeto que no le pertenece, pida al propietario del objeto que genere una URL prefirmada que le permita descargar el objeto. Para obtener más información, consulte the section called “Descarga de un objeto de otra Cuenta de AWS”.

Cuando descarga objetos fuera de la red de AWS, se aplican tarifas de transferencia de datos. La transferencia de datos dentro de la red de AWS es gratuita si está en la misma Región de AWS, pero se le cobrará por cualquier solicitud `GET`. Para obtener más información sobre los costes de transferencia de datos y los cargos por recuperación de datos, consulte los precios de Amazon S3.

Temas

- Descarga de un objeto

Descarga de un objeto

Puede descargar un objeto utilizando la consola de Amazon S3, AWS CLI, los SDK de AWS o la API de REST.

Uso de la consola de S3

En esta sección se explica cómo utilizar la consola de Amazon S3 para descargar un objeto de un bucket de S3.

> **Note**
>
> - Solo se puede descargar un objeto a la vez.
> - Si utiliza la consola de Amazon S3 para descargar un objeto cuyo nombre de clave termine con un punto (`.`), se eliminará el punto del nombre de clave del objeto descargado. Para conservar el punto al final del nombre del objeto descargado, debe usar la AWS Command Line Interface (AWS CLI), los SDK de AWS o la API de REST de Amazon S3.

Para descargar un objeto desde un bucket de S3

1. Inicie sesión en la AWS Management Console y abra la consola de Amazon S3 en https://console.aws.amazon.com/s3/.
2. En la lista Buckets (Buckets), elija el nombre del bucket que contiene el objeto que desea descargar.
3. Puede descargar un objeto de un bucket de S3 de cualquiera de las siguientes maneras:
   - Seleccione la casilla de verificación situada junto al objeto y elija Descargar. Si desea descargar el objeto a una carpeta específica, en el menú Acciones, seleccione Descargar como.

- Si desea descargar una versión específica del objeto, active Mostrar versiones (situado junto al cuadro de búsqueda). Seleccione la casilla de verificación situada junto a la versión del objeto que desee y elija Descargar. Si desea descargar el objeto a una carpeta específica, en el menú Acciones, seleccione Descargar como.

Uso de la AWS CLI

En el siguiente comando de ejemplo de `get-object`, se muestra cómo puede utilizar la AWS CLI para descargar un objeto de Amazon S3. Este comando obtiene el objeto *`folder/my_image`* del bucket *`amzn-s3-demo-bucket1`*. El objeto se descargará en un archivo denominado *`my_downloaded_image`*.

```
aws s3api get-object --bucket amzn-s3-demo-bucket1 --key folder/
my_image my_downloaded_image
```

Para obtener más información y ejemplos, consulte get-object en la referencia de comandos de AWS CLI.

Uso de los AWS SDK

Para ver ejemplos de cómo descargar un objeto con los SDK de AWS, consulte Uso de `GetObject` con un AWS SDK o la CLI.

Para obtener información general sobre el uso de diferentes SDK de AWS, consulte Desarrollo con Amazon S3 mediante los SDK de AWS.

Uso de la API de REST

Puede utilizar la API de REST para recuperar objetos de Amazon S3. Para obtener más información, consulte GetObject en la Referencia de la API de Amazon Simple Storage Service.

Descarga de varios objetos

Puede descargar varios objetos mediante AWS CloudShell, la AWS CLI o los SDK de AWS.

Uso de AWS CloudShell en AWS Management Console

AWS CloudShell es un intérprete de comandos previamente autenticado y basado en el navegador, que se puede lanzar directamente desde la página web de la AWS Management Console.

Para obtener más información acerca de AWS CloudShell, consulte ¿Qué es CloudShell? en la Guía del usuario de AWS CloudShell.

> **Important**
>
> Con AWS CloudShell, su directorio principal tiene capacidad de almacenamiento de hasta 1 GB por Región de AWS Por lo tanto, no puede sincronizar buckets con objetos que superen en total esta cantidad. Para ver más limitaciones, consulta las Service quotas and restrictions (Cuotas y restricciones del servicio) en la Guía del usuario de AWS CloudShell.

Para descargar objetos mediante AWS CloudShell

1. Inicie sesión en la AWS Management Console y abra la consola de CloudShell en https://console.aws.amazon.com/cloudshell/.
2. Ejecute el siguientes comando para sincronizar objetos del bucket con CloudShell. El siguiente comando sincroniza los objetos del bucket denominado *amzn-s3-demo-bucket1* y crea una carpeta denominada *temp* en CloudShell. CloudShell sincroniza los objetos con esta carpeta. Para usar este comando, sustituya *user input placeholders* por su información.

   ```
   aws s3 sync s3://amzn-s3-demo-bucket1 ./temp
   ```

   > **Note**
   >
   > Para realizar una coincidencia de patrones para excluir o incluir objetos concretos, puede utilizar los parámetros `--exclude "value"` y `--include "value"` con el comando `sync`.

3. Ejecute el siguiente comando para comprimir los objetos en la carpeta con el nombre *temp*en un archivo denominado *temp.zip*.

   ```
   zip temp.zip -r temp/
   ```

4. Elija Acciones y luego Descargar archivo.
5. Escriba el nombre de archivo **temp.zip** y luego elija Descargar.

6. (Opcional) Elimine el archivo *temp.zip* y los objetos que están sincronizados con la carpeta *temp* de CloudShell. Con AWS CloudShell, tiene un almacenamiento persistente de hasta 1 GB para cada Región de AWS.

   Los siguientes comandos de ejemplo se pueden utilizar para eliminar el archivo `.zip` y la carpeta. Para utilizar este comando de ejemplo, sustituya *user input placeholders* por su propia información.

   ```
   rm temp.zip && rm -rf temp/
   ```

Uso de la AWS CLI

Los siguientes ejemplos muestran cómo se puede utilizar AWS CLI para descargar todos los archivos u objetos que tienen el directorio o prefijo especificados. Este comando copia todos los objetos del bucket *amzn-s3-demo-bucket1* a su directorio actual. Para usar este comando de ejemplo, use el nombre de su bucket en lugar de *amzn-s3-demo-bucket1*.

```
aws s3 cp s3://amzn-s3-demo-bucket1 . --recursive
```

El siguiente comando descarga todos los objetos con el prefijo *logs* del bucket *amzn-s3-demo-bucket1* en su directorio actual. También usa los parámetros `--exclude` y `--include` para copiar solo los objetos con el sufijo *.log*. Para utilizar este comando de ejemplo, sustituya *user input placeholders* por su propia información.

```
aws s3 cp s3://amzn-s3-demo-bucket1/logs/ . --recursive --exclude "*" --include "*.log"
```

Para obtener más información y ejemplos, consulte cp en la referencia de comandos de AWS CLI.

Uso de los AWS SDK

Para ver ejemplos de cómo descargar todos los objetos de un bucket de Amazon S3 con los SDK de AWS, consulte Descargar todos los objetos de un bucket de Amazon Simple Storage Service (Amazon S3) en un directorio local.

Para obtener información general sobre el uso de diferentes SDK de AWS, consulte Desarrollo con Amazon S3 mediante los SDK de AWS.

Para descargar parte de un objeto

Puede descargar parte de un objeto mediante la AWS CLI o la API de REST. Para ello, utilice parámetros adicionales para especificar qué parte de un objeto desea descargar.

Uso de la AWS CLI

El siguiente comando de ejemplo ejecuta una solicitud `GET` de un rango de bytes en el objeto denominado *`folder/my_data`* del bucket denominado *`amzn-s3-demo-bucket1`*. En la solicitud, el rango de bytes debe llevar el prefijo `bytes=`. El objeto parcial se descarga en el archivo de salida denominado *`my_data_range`*. Para utilizar este comando de ejemplo, sustituya *`user input placeholders`* por su propia información.

```
aws s3api get-object --bucket amzn-s3-demo-bucket1 --key folder/my_data --range
 bytes=0-500 my_data_range
```

Para obtener más información y ejemplos, consulte get-object en la referencia de comandos de AWS CLI.

Para más información sobre el encabezado `Range` HTTP, consulte el documento RFC 9110 en la página web del editor de RFC.

> **Note**
>
> Amazon S3 no admite la recuperación de varios rangos de datos en una sola solicitud `GET`.

Uso de la API de REST

Puede usar los parámetros `partNumber` y `Range` de la API de REST para recuperar partes de objetos de Amazon S3. Para obtener más información, consulte GetObject en la Referencia de la API de Amazon Simple Storage Service.

Descarga de un objeto de otra Cuenta de AWS

Puede utilizar una URL prefirnada para conceder a otras personas un acceso de tiempo limitado a sus objetos sin necesidad de actualizar su política de buckets.

La URL prefirmada puede introducirse en un navegador o utilizarse por un programa para descargar un objeto. Las credenciales que utiliza la URL son las del usuario de AWS que generó la URL.

Una vez creada la URL, cualquier persona que tenga la URL prefirmada puede descargar el objeto correspondiente hasta que la URL caduque.

Uso de una URL prefirmada en la consola S3

Puede utilizar la consola de Amazon S3 para generar una URL prefirmada para compartir un objeto si sigue estos pasos. Al utilizar la consola, el tiempo máximo de caducidad de una URL prefirmada es de 12 horas desde el momento de su creación.

Para generar una URL prefirmada con la consola de Amazon S3

1. Inicie sesión AWS Management Console Management Console y abra la consola de Amazon S3 en https://console.aws.amazon.com/s3/.
2. En el panel de navegación izquierdo, elija Instancias.
3. En la lista Buckets (Buckets), elija el nombre del bucket que contiene el objeto para el que desea generar una URL prefirmada.
4. En la lista Objects (Objetos), seleccione el objeto para el que desea crear una URL prefirmada.
5. En el menú Acciones, elija Compartir con una URL prefirmada.
6. Especifique cuánto tiempo de validez desea que tenga la URL prefirmada.
7. Elija Create presigned URL (Crear URL prefirmada).
8. Cuando aparece un mensaje de confirmación, la URL se copia automáticamente en el portapapeles. Verá un botón para copiar la URL prefirmada en caso de que necesite volver a copiarla.
9. Para descargar el objeto, pegue la URL en cualquier navegador y el objeto intentará descargarse.

Para obtener más información sobre las direcciones URL prefirmadas y otros métodos para crearlas, consulte Uso de URL prefirmadas.

Descarga de registros archivados

Para reducir los costos de almacenamiento de los objetos a los que se accede con poca frecuencia, puede archivar dichos objetos. Cuando archiva un objeto, se traslada a un almacenamiento de bajo costo, lo que significa que no puede acceder a él en tiempo real. Para descargar un objeto archivado, primero debe restaurarlo.

Puede restaurar objetos que se han archivado en minutos o en horas, según la clase de almacenamiento. Puede restaurar un objeto archivado mediante la consola de Amazon S3, Operaciones por lotes de S3, la API de REST de Amazon S3, los SDK de AWS y la AWS Command Line Interface (AWS CLI).

Para obtener instrucciones, consulte Restauración de un objeto archivado. Tras restaurar el objeto archivado, puede descargarlo.

Solución de problemas al descargar objetos

Los permisos insuficientes o las políticas de usuario AWS Identity and Access Management (IAM) o bucket incorrectas pueden provocar errores al intentar descargar objetos de Amazon S3. Estos problemas suelen provocar errores de acceso denegado (403 prohibido), en los que Amazon S3 no puede permitir el acceso a un recurso.

Para conocer las causas habituales de los errores de acceso denegado (403 prohibido), consulte Solucionar errores de acceso denegado (403 Prohibido) en Amazon S3.

Comprobación de la integridad de objetos

Amazon S3 utiliza valores de suma de comprobación para verificar la integridad de los datos que carga o descarga de Amazon S3. Además, puede solicitar que se calcule otro valor de suma de comprobación para cualquier objeto que almacene en Amazon S3. Puede seleccionar uno de los varios algoritmos de suma de comprobación que se utilizarán al cargar o copiar los datos. Amazon S3 utiliza este algoritmo para calcular un valor de suma de comprobación adicional y almacenarlo como parte de los metadatos del objeto. Para obtener más información sobre cómo utilizar sumas de comprobación adicionales para verificar la integridad de los datos, consulte el Tutorial: Checking the integrity of data in Amazon S3 with additional checksums (Comprobación de la integridad de los datos en Amazon S3 con sumas de comprobación adicionales).

Al cargar un objeto, puede incluir opcionalmente una suma de comprobación precalculada como parte de la solicitud. Amazon S3 compara la suma de comprobación proporcionada con la suma de comprobación que calcula mediante el algoritmo especificado. Si los valores no coinciden, Amazon S3 genera un error.

Uso de algoritmos de suma de comprobación admitidos

Amazon S3 le ofrece la opción de elegir el algoritmo de suma de comprobación que se utiliza para validar los datos durante la carga o descarga. Puede seleccionar uno de los siguientes algoritmos

de comprobación de integridad de datos Secure Hash Algoritms (SHA) o Cyclic Redundancy Check (CRC):

- CRC32
- CRC 32C
- SHA-1
- SHA-256

Al cargar un objeto, puede especificar el algoritmo que desea utilizar:

- Cuando utilice el campo AWS Management Console, seleccione el algoritmo de suma de comprobación que desee utilizar. Cuando lo haga, puede especificar opcionalmente el valor de la suma de comprobación del objeto. Cuando Amazon S3 recibe el objeto, calcula la suma de comprobación mediante el algoritmo especificado. Si los valores de la suma de comprobación no coinciden, Amazon S3 genera un error.
- Cuando se utiliza un SDK, se puede establecer el valor del parámetro `x-amz-sdk-checksum-algorithm` en el algoritmo que desea que Amazon S3 utilice al calcular la suma de comprobación. Amazon S3 calcula automáticamente el valor de la suma de comprobación.
- Si utiliza la API de REST, no utilice el parámetro `x-amz-sdk-checksum-algorithm`. En su lugar, utilice uno de los encabezados específicos del algoritmo (por ejemplo, `x-amz-checksum-crc32`).

Para obtener más información acerca de la carga de objetos, consulte Carga de objetos.

Para aplicar cualquiera de estos valores de suma de comprobación a objetos que ya se han cargado en Amazon S3, puede copiar el objeto. Al copiar un objeto, puede especificar si desea utilizar el algoritmo de suma de comprobación existente o utilizar uno nuevo. Puede especificar un algoritmo de suma de comprobación cuando utilice cualquier mecanismo admitido para copiar objetos, incluidas las operaciones por lotes de S3. Para obtener más información sobre la herramienta de operaciones por lotes de S3, consulte Realización de operaciones por lotes a gran escala en objetos de Amazon S3.

> **Important**
>
> Si utiliza una carga multiparte con sumas de comprobación adicionales, los números de partes multiparte deben ser consecutivos. Al utilizar sumas de comprobación adicionales, si

intenta completar una solicitud de carga multiparte con números de parte no consecutivos, Amazon S3 genera un error `500 Internal Server Error` de HTTP.

Después de cargar objetos, puede obtener el valor de la suma de comprobación y compararlo con un valor de suma de comprobación precalculado o almacenado previamente que se ha calculado utilizando el mismo algoritmo.

Uso de la consola de S3

Para obtener más información sobre el uso de la consola y la especificación de los algoritmos de suma de comprobación que se utilizan al cargar objetos, consulte Carga de objetos y el Tutorial: Comprobación de la integridad de los datos en Amazon S3 con sumas de comprobación adicionales.

Uso de los SDK de AWS

En el siguiente ejemplo, se muestra cómo se pueden utilizar los SDK de AWS para cargar un archivo grande con carga multiparte, descargar un archivo grande y validar un archivo de carga multiparte, todo con SHA-256 para la validación de archivos.

Java

Example Ejemplo: carga, descarga y verificación de un archivo grande con SHA-256

Para obtener instrucciones sobre cómo crear y probar una muestra funcional, consulte Introducción en la Guía del desarrollador de AWS SDK for Java.

```
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();
            }
        }
    }
```

Uso de la API de REST

Puede enviar solicitudes REST para cargar un objeto con un valor de suma de comprobación para verificar la integridad de los datos con PutObject. También puede recuperar el valor de suma de comprobación de los objetos con GetObject o HeadObject.

Uso de la AWS CLI

Envíe una solicitud PUT para cargar un objeto de hasta 5 GB en una única operación. Para obtener más información, consulte `PutObject` en la Referencia de comandos de AWS CLI. También puede

utilizar `get-object` y `head-object` para recuperar la suma de comprobación de un objeto ya cargado para verificar la integridad de los datos.

Para ver más información, consulte Amazon S3 CLI FAQ en la Guía del usuario de la AWS Command Line Interface.

Uso de Content-MD5 al cargar objetos

Otra forma de verificar la integridad del objeto después de cargarlo es proporcionar un resumen MD5 del objeto al cargarlo. Si calcula el resumen MD5 de su objeto, puede proporcionar el resumen con el comando PUT mediante el encabezado `Content-MD5`.

Tras cargar el objeto, Amazon S3 calcula el resumen MD5 del objeto y lo compara con el valor que proporcionó. La solicitud se realiza correctamente solo si los dos resúmenes coinciden.

No es necesario suministrar un resumen MD5, pero puede usarlo para verificar la integridad del objeto como parte del proceso de carga.

Uso de Content-MD5 y ETag para verificar los objetos cargados

La etiqueta de entidad (ETag) de un objeto representa una versión específica de ese objeto. Tenga en cuenta que la ETag solo refleja los cambios en su contenido, no en los metadatos. Si solo cambian los metadatos de un objeto, la ETag sigue siendo la misma.

Según el objeto, la ETag del objeto puede ser un resumen MD5 de los datos del objeto:

- Si un objeto se ha creado con la operación `PutObject`, `PostObject` o `CopyObject`, o a través de AWS Management Console, y también tiene texto sin formato o está cifrado mediante cifrado del lado del servidor con claves administradas de Amazon S3 (SSE-S3), ese objeto tiene una ETag que es un resumen MD5 de sus datos de objeto.
- Si un objeto se ha creado con la operación `PutObject`, `PostObject` o `CopyObject`, o a través de AWS Management Console, y está cifrado mediante cifrado del lado del servidor con claves proporcionadas por el cliente (SSE-C) o mediante cifrado del lado del servidor con AWS Key Management Service (AWS KMS) (SSE-KMS), ese objeto tiene una ETag que no es un resumen MD5 de sus datos de objeto.
- Si un objeto se crea mediante la operación `Multipart Upload` o `Part Copy`, la ETag del objeto no es un resumen MD5, independientemente del método de cifrado. Si un objeto tiene más de 16 MB, la AWS Management Console carga o copia ese objeto como carga multiparte y, por lo tanto, la ETag no es un resumen MD5.

Para objetos en los que la ETag es el resumen `Content-MD5` del objeto, puede comparar el valor de ETag del objeto con un resumen `Content-MD5` calculado o almacenado previamente.

Uso de sumas de comprobación finales

Al cargar objetos en Amazon S3, puede proporcionar una suma de comprobación calculada previamente para el objeto o utilizar un SDK de AWS para crear automáticamente sumas de comprobación finales en su nombre. Si decide utilizar una suma de comprobación final, Amazon S3 genera automáticamente la suma de comprobación mediante el algoritmo especificado y la utiliza para validar la integridad del objeto durante la carga.

Para crear una suma de comprobación final cuando utilice un SDK de AWS, rellene el parámetro `ChecksumAlgorithm` con su algoritmo preferido. El SDK utiliza ese algoritmo para calcular la suma de comprobación de su objeto (o partes de objeto) y lo anexa automáticamente al final de la solicitud de carga. Este comportamiento le ahorra tiempo porque Amazon S3 realiza la verificación y la carga de sus datos en un solo pase.

> **Important**
>
> Si utiliza S3 Object Lambda, todas las solicitudes a S3 Object Lambda se firman mediante `s3-object-lambda` en lugar de `s3`. Este comportamiento afecta a la firma de los valores de suma de comprobación final. Para obtener más información acerca de S3 Object Lambda, consulte Transformación de objetos con Lambda para objetos S3.

Uso de sumas de comprobación a nivel de parte para cargas multiparte

Cuando los objetos se cargan en Amazon S3, se pueden cargar como un solo objeto o mediante el proceso de carga multiparte. Los objetos de más de 16 MB que se cargan a través de la consola se cargan automáticamente mediante cargas multiparte. Para obtener más información acerca de las cargas multipartes, consulte Carga y copia de objetos con la carga multiparte.

Cuando un objeto se carga como carga multiparte, la ETag del objeto no es un resumen MD5 de todo el objeto. Amazon S3 calcula el resumen MD5 de cada parte individual a medida que se carga. Los resúmenes MD5 se utilizan para determinar la ETag del objeto final. Amazon S3 concatena los bytes de los resúmenes MD5 y, a continuación, calcula el resumen MD5 de estos valores concatenados. El último paso para crear la ETag es cuando Amazon S3 añade un guion con el número total de partes al final.

Por ejemplo, supongamos que tenemos un objeto cargado con una carga multiparte que tiene una ETag de `C9A5A6878D97B48CC965C1E41859F034-14`. En este caso, `C9A5A6878D97B48CC965C1E41859F034` es el resumen MD5 de todos los resúmenes concatenados. El `-14` indica que hay 14 artículos asociados a la carga multiparte de este objeto.

Si ha habilitado valores de suma de comprobación adicionales para el objeto multiparte, Amazon S3 calcula la suma de comprobación de cada parte individual mediante el algoritmo de suma de comprobación especificado. La suma de comprobación del objeto completado se calcula de la misma manera que Amazon S3 calcula el resumen MD5 para la carga multiparte. Puede utilizar esta suma de comprobación para verificar la integridad del objeto.

Para recuperar información sobre el objeto, incluida la cantidad de partes que componen todo el objeto, puede utilizar la operación GetObjectAttributes. Con sumas de comprobación adicionales, también puede recuperar información de cada parte individual que incluye el valor de suma de comprobación de cada parte.

En el caso de cargas completadas, puede obtener la suma de comprobación de una parte individual mediante la operación GetObject o HeadObject y especificar un número de parte o rango de bytes que se ajuste a una sola parte. Si quiere recuperar los valores de la suma de comprobación de partes individuales de las cargas de varias partes que aún están en curso, puede utilizar ListParts.

Debido a la forma en que Amazon S3 calcula la suma de comprobación de objetos multiparte, es posible que el valor de la suma de comprobación del objeto cambie si lo copia. Si utiliza un SDK o la API de REST y llama a CopyObject, Amazon S3 copia cualquier objeto hasta el tamaño límite de la operación API `CopyObject`. Amazon S3 realiza esta copia como una sola acción, independientemente de si el objeto se ha cargado en una sola solicitud o como parte de una carga multiparte. Con un comando de copia, la suma de comprobación del objeto es una suma de comprobación directa de todo el objeto. Si el objeto se subió originalmente mediante una carga multiparte, el valor de la suma de comprobación cambia aunque los datos no lo hayan hecho.

> **Note**
>
> Los objetos que superen los límites de tamaño de la operación de API `CopyObject` deben utilizar comandos de copia multiparte.

> **Important**
>
> Cuando realiza algunas operaciones con la AWS Management Console, Amazon S3 utiliza una carga multiparte si el objeto tiene un tamaño superior a 16 MB. En este caso, la suma de comprobación no es una suma de comprobación directa del objeto completo, sino un cálculo basado en los valores de suma de comprobación de cada parte individual.
> Por ejemplo, supongamos que tenemos un objeto de 100 MB que ha subido como carga directa de una sola parte mediante la API de REST. La suma de comprobación en este caso es una suma de comprobación de todo el objeto. Si posteriormente utiliza la consola para cambiar el nombre de ese objeto, copiarlo, cambiar la clase de almacenamiento o editar los metadatos, Amazon S3 utiliza la funcionalidad de carga multiparte para actualizar el objeto. Como resultado, Amazon S3 crea un nuevo valor de suma de comprobación para el objeto que se calcula en función de los valores de suma de comprobación de las partes individuales.
> La lista anterior de operaciones de la consola no es una lista completa de todas las acciones que se pueden realizar en la AWS Management Console que hacen que Amazon S3 actualice el objeto mediante la funcionalidad de carga multiparte. Tenga en cuenta que siempre que utilice la consola para actuar sobre objetos de más de 16 MB, es posible que el valor de la suma de comprobación no sea la suma de comprobación de todo el objeto.

Eliminación de objetos de Amazon S3

Puede eliminar uno o más objetos directamente de Amazon S3 mediante la consola de Amazon S3, los SDK de AWS, AWS Command Line Interface (AWS CLI) o la API de REST. Debido a que todos los objetos en el bucket de S3 generan costos de almacenamiento, debe eliminar los objetos cuando ya no los necesita. Por ejemplo, si recopila archivos de registro, es recomendable eliminarlos cuando ya no sean necesarios. También puede configurar una regla de ciclo de vida para eliminar los objetos, como los archivos de registro, de manera automática. Para obtener más información, consulte the section called “Configurar el ciclo de vida”.

Para obtener información sobre las características y precios de Amazon S3, consulte Precios de Amazon S3.

Tiene las siguientes opciones de API cuando elimina un objeto:

- Eliminar un solo objeto: Amazon S3 proporciona la API `DELETE` (`DeleteObject`) que puede utilizar para eliminar un objeto en una sola solicitud HTTP.

- Eliminar varios objetos : Amazon S3 proporciona la operación de la API (`DeleteObjects`) de eliminación de varios objetos que puede usar para eliminar hasta 1000 objetos en una sola solicitud HTTP.

Cuando elimina objetos de un bucket que no tiene habilitado el control de versiones, solo se proporciona el nombre de la clave de objeto. Sin embargo, cuando elimina objetos de un bucket que tiene el control de versiones habilitado, puede proporcionar opcionalmente el ID de versión del objeto a fin de eliminar una versión específica del objeto.

Eliminación de objetos mediante programación de un bucket que tiene habilitado el control de versiones

Si el bucket tiene habilitado el control de versiones, pueden existir varias versiones del mismo objeto en el bucket. Cuando trabaja con buckets que tienen habilitado el control de versiones, las operaciones de la API de eliminación permiten las siguientes opciones:

- Especificar una solicitud de eliminación sin versión: especifique solo la clave del objeto y no el ID de versión. En este caso, Amazon S3 crea un marcador de eliminación y devuelve su ID de versión en la respuesta. Esto hace que el objeto desaparezca del bucket. Para obtener información acerca del control de versiones de los objetos y el concepto de marcador de eliminación, consulte Usar el control de versiones en buckets de S3.
- Especificar una solicitud de eliminación con versión: especifique la clave y también un ID de versión. En este caso, se obtienen los siguientes dos resultados posibles:
  - Si el ID de versión se asigna a una versión de objeto específica, Amazon S3 elimina la versión específica del objeto.
  - Si el ID de versión se asigna al marcador de eliminación de ese objeto, Amazon S3 elimina el marcador de eliminación. Esto hace que el objeto aparezca nuevamente en su bucket.

Eliminar objetos de un bucket habilitado para la MFA

Cuando elimine objetos de un bucket habilitado para la autenticación multifactor (MFA), tenga en cuenta lo siguiente:

- Si proporciona un token de MFA no válido, la solicitud siempre devuelve un error.
- Si tiene un bucket habilitado para la MFA y realiza una solicitud de eliminación de objetos con control de versiones (proporciona una clave y un ID de versión del objeto), la solicitud dará

error si no proporciona un token de MFA válido. Además, cuando usa la operación de la API de eliminación de varios objetos en un bucket habilitado para la MFA, si alguna de las eliminaciones es una solicitud de eliminación con control de versiones (es decir, especifica una clave y un ID de versión del objeto), habrá un error en toda la solicitud si no proporciona un token de MFA.

Sin embargo, en los siguientes casos, la solicitud se realiza correctamente:

- Si tiene un bucket habilitado para la MFA y realiza una solicitud de eliminación sin control de versiones (no elimina un objeto con control de versiones), y no proporciona un token de MFA, la eliminación se realiza con éxito.
- Si tiene una solicitud de eliminación de varios objetos en la que se especifica que solo se deben eliminar objetos sin control de versiones de un bucket habilitado para la MFA, y no proporciona un token de MFA, las eliminaciones se realizan correctamente.

Para obtener información acerca de la eliminación de MFA, consulte Configurar la eliminación de MFA.

Temas

Eliminación de un solo objeto

Puede utilizar la consola de Amazon S3 o la API DELETE para eliminar un solo objeto existente de un bucket de S3. Para obtener más información sobre la eliminación de objetos en Amazon S3, consulte Eliminación de objetos de Amazon S3.

Debido a que todos los objetos en el bucket de S3 generan costos de almacenamiento, debe eliminar los objetos cuando ya no los necesita. Por ejemplo, si recopila archivos de registro, es recomendable eliminarlos cuando ya no sean necesarios. También puede configurar una regla de ciclo de vida para eliminar los objetos, como los archivos de registro, de manera automática. Para obtener más información, consulte the section called “Configurar el ciclo de vida”.

Para obtener información sobre las características y precios de Amazon S3, consulte Precios de Amazon S3.

Uso de la consola de S3

Siga estos pasos para utilizar la consola de Amazon S3 a fin de eliminar un solo objeto de un bucket.

> **Warning**
>
> Si elimina permanentemente un objeto o una versión específica del objeto en la consola de Amazon S3, la eliminación no se puede deshacer.

Cómo eliminar un objeto que tiene el control de versiones activado o suspendido

> **Note**
>
> Si el ID de versión de un objeto de un bucket con control de versiones suspendido está marcado como NULL, S3 elimina el objeto de forma permanente, ya que no existen versiones anteriores. Sin embargo, si aparece un ID de versión válido para el objeto en un bucket con control de versiones suspendido, S3 crea un marcador de eliminación para el objeto eliminado y, al mismo tiempo, conserva las versiones anteriores del objeto.

1. Inicie sesión en la AWS Management Console y abra la consola de Amazon S3 en https://console.aws.amazon.com/s3/.
2. En la lista Bucket name (Nombre del bucket), seleccione el nombre del bucket que contiene el objeto que desea eliminar.
3. Seleccione el objeto y, a continuación, elija Eliminar.
4. Para confirmar la eliminación de la lista de objetos en Objetos especificados en el cuadro de texto ¿Eliminar objetos?, introduzca **`delete`**.

Cómo eliminar una versión específica del objeto en un bucket con control de versiones de forma definitiva

> **Warning**
>
> Si elimina una versión específica del objeto de forma permanente en Amazon S3, la eliminación no se puede deshacer.

1. Inicie sesión en la AWS Management Console y abra la consola de Amazon S3 en https://console.aws.amazon.com/s3/.
2. En la lista Bucket name (Nombre del bucket), seleccione el nombre del bucket que contiene el objeto que desea eliminar.
3. Seleccione los objetos que desea eliminar.
4. Elija Mostrar versiones.
5. Seleccione la versión del objeto y, a continuación, elija Eliminar.
6. Para confirmar la eliminación definitiva de la versión del objeto enumerada en Objetos especificados en el cuadro de texto ¿Eliminar objetos?, introduzca Eliminar definitivamente. Amazon S3 eliminará la versión del objeto específica de forma definitiva.

Cómo eliminar un objeto de forma definitiva de un bucket de Amazon S3 que no tiene activado el control de versiones

> **Warning**
>
> Si elimina un objeto de forma definitiva en Amazon S3, la eliminación no se puede deshacer. Además, en el caso de los buckets que no tengan el control de versiones activado, las eliminaciones son definitivas.

1. Inicie sesión en la AWS Management Console y abra la consola de Amazon S3 en https://console.aws.amazon.com/s3/.
2. En la lista Bucket name (Nombre del bucket), seleccione el nombre del bucket que contiene el objeto que desea eliminar.
3. Seleccione el objeto y, a continuación, elija Eliminar.
4. Para confirmar la eliminación definitiva objeto enumerado en Objetos especificados en el cuadro de texto ¿Eliminar objetos?, introduzca Eliminar definitivamente.

> **Note**
>
> Si tiene problemas para eliminar el objeto, consulte Quiero eliminar los objetos versionados de forma permanente.

Uso de los AWS SDK

En los ejemplos siguientes se muestra cómo puede utilizar los AWS SDK para eliminar un objeto de un bucket. Para obtener más información, consulte DELETE Object en la Referencia de API de Amazon Simple Storage Service

Si el bucket tiene habilitado Control de versiones de S3, tiene las siguientes opciones:

- Eliminar una versión específica del objeto especificando un ID de versión.
- Eliminar un objeto sin especificar un ID de versión. En ese caso, Amazon S3 agrega un marcador de eliminación al objeto.

Para obtener más información sobre el control de versiones de S3, consulte Usar el control de versiones en buckets de S3.

Java

Example Ejemplo 1: eliminar un objeto (bucket sin control de versiones)

En el siguiente ejemplo se asume que el bucket no tiene habilitado el control de versiones y que el objeto no tiene ID de versión. En la solicitud de eliminación, especificará solo la clave del objeto y no el ID de versión.

Para obtener instrucciones sobre cómo crear y probar una muestra funcional, consulte Introducción en la Guía del desarrollador de AWS SDK for Java.

```
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 Ejemplo 2: eliminar un objeto (bucket con control de versiones)

En el siguiente ejemplo se elimina un objeto de un bucket con control de versiones. En el ejemplo se elimina una versión del objeto concreta especificando el nombre de clave y el ID de versión del objeto.

En el ejemplo se realiza lo siguiente:

1. Añadir un objeto de muestra al bucket. Amazon S3 devuelve un ID de versión del objeto recién añadido. El ejemplo utiliza este ID de versión en la solicitud de eliminación.
2. Eliminar la versión del objeto especificando tanto el nombre de clave como el ID de versión del objeto. Si no hay otras versiones de ese objeto, Amazon S3 elimina el objeto en su totalidad. De lo contrario, Amazon S3 solo elimina la versión especificada.

> **Note**
>
> Puede obtener ID de versiones de un objeto con la solicitud `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

En el siguiente ejemplo se muestra cómo eliminar un objeto tanto de un bucket con control de versiones como de un bucket sin control versiones. Para obtener más información sobre el control de versiones de S3, consulte Usar el control de versiones en buckets de S3.

Example Eliminar un objeto de un bucket sin control de versiones

En el siguiente ejemplo de código C# se elimina un objeto de un bucket sin control de versiones. En el ejemplo se supone que los objetos no tienen ID de versión, por lo que no tiene que especificar los ID de versión. Especifique solo la clave del objeto.

Para obtener información acerca de cómo configurar y ejecutar ejemplos de código, consulte Introducción al SDK de AWS para .NET en la Guía para desarrolladores del SDK de AWS para .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 Eliminar un objeto de un bucket con control de versiones

En el siguiente ejemplo de código C# se elimina un objeto de un bucket con control de versiones. Se elimina una versión de objeto concreta especificando el nombre de clave y el ID de versión del objeto.

El código realiza las siguientes tareas:

1. Habilita el control de versiones de S3 en un bucket que especifique (si el control de versiones de S3 ya está habilitado, esto no tiene efecto).
2. Añadir un objeto de muestra al bucket. Como respuesta, Amazon S3 devuelve un ID de versión del nuevo objeto añadido. El ejemplo utiliza este ID de versión en la solicitud de eliminación.
3. Elimina el objeto de muestra especificando tanto el nombre de clave como el ID de versión del objeto.

> **Note**
>
> También puede obtener el ID de versión de un objeto con la solicitud `ListVersions`.
>
> ```
> var listResponse = client.ListVersions(new ListVersionsRequest { BucketName
>  = bucketName, Prefix = keyName });
> ```

Para obtener información acerca de cómo configurar y ejecutar ejemplos de código, consulte Introducción al SDK de AWS para .NET en la Guía para desarrolladores del SDK de AWS para .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

En este ejemplo, se muestra cómo utilizar las clases de la versión 3 de AWS SDK for PHP a fin de eliminar un objeto de un bucket sin control de versiones. Para obtener información acerca de la eliminación de un objeto de un bucket con control de versiones, consulte Uso de la API de REST.

Para obtener más información acerca de la API del SDK de AWS para Ruby, consulte SDK de AWS para Ruby, versión 2.

En el siguiente ejemplo de PHP se elimina un objeto de un bucket. Como este ejemplo muestra cómo eliminar objetos de buckets sin control de versiones, proporciona solo el nombre del bucket y la clave del objeto (no un ID de versión) en la solicitud de eliminación.

```
<?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();
```

Uso de la AWS CLI

Para eliminar un objeto por solicitud, utilice la API `DELETE`. Para obtener más información, consulte DELETE Object. Para obtener más información acerca del uso de la CLI para eliminar un objeto, vea delete-object.

Uso de la API de REST

Puede utilizar los AWS SDK para eliminar un objeto. Sin embargo, si su aplicación lo requiere, puede enviar solicitudes REST directamente. Para obtener más información, consulte DELETE Object en la Referencia de API de Amazon Simple Storage Service.

Eliminación de varios objetos

Debido a que todos los objetos en el bucket de S3 generan costos de almacenamiento, debe eliminar los objetos cuando ya no los necesita. Por ejemplo, si recopila archivos de registro, es recomendable eliminarlos cuando ya no sean necesarios. También puede configurar una regla de ciclo de vida para eliminar los objetos, como los archivos de registro, de manera automática. Para obtener más información, consulte the section called “Configurar el ciclo de vida”.

Para obtener información sobre las características y precios de Amazon S3, consulte Precios de Amazon S3.

Puede utilizar la consola de Amazon S3, los AWS SDK o la API de REST para eliminar varios objetos de forma simultánea de un bucket de S3.

Uso de la consola de S3

Siga estos pasos para utilizar la consola de Amazon S3 a fin de eliminar varios objetos de un bucket.

Warning

- No se puede deshacer la eliminación de un objeto especificado.
- Esta acción elimina todos los objetos especificados. Al eliminar carpetas, espere a que finalice la acción de eliminación antes de agregar nuevos objetos a la carpeta. De lo contrario, es posible que también se eliminen objetos nuevos.
- Cuando se eliminan objetos de un bucket sin el control de versiones habilitado, Amazon S3 elimina de forma permanente los objetos.
- Cuando se eliminan objetos de un bucket con el control de versiones de bucket habilitado o suspendido, Amazon S3 crea marcadores de eliminación. Para obtener más información, consulte Trabajar con marcadores de eliminación.

Eliminación de un objeto que tiene el control de versiones habilitado o suspendido

Note

Si los ID de versión del objeto de un bucket con control de versiones suspendido están marcados como NULL, S3 elimina los objetos de forma permanente, ya que no existen versiones anteriores. Sin embargo, si aparece un ID de versión válido para los objetos en un bucket con control de versiones suspendido, S3 crea los marcadores de eliminación para los objetos eliminados y, al mismo tiempo, conserva las versiones anteriores de los objetos.

1. Inicie sesión AWS Management Console Management Console y abra la consola de Amazon S3 en https://console.aws.amazon.com/s3/.
2. En la lista Nombre del bucket, seleccione el nombre del bucket del que desea eliminar los objetos.
3. Seleccione los objetos y, a continuación, elija Eliminar.
4. Para confirmar la eliminación de la lista de objetos en Objetos especificados en el cuadro de texto ¿Eliminar objetos?, introduzca **`delete`**.

Eliminación permanente de versiones específicas de objetos en un bucket con control de versiones habilitado

> **Warning**
>
> Cuando elimina de forma permanente versiones específicas de objetos en Amazon S3, la eliminación no se puede deshacer.

1. Inicie sesión AWS Management Console Management Console y abra la consola de Amazon S3 en https://console.aws.amazon.com/s3/.
2. En la lista Nombre del bucket, seleccione el nombre del bucket del que desea eliminar los objetos.
3. Seleccione los objetos de que desea eliminar.
4. Elija Mostrar versiones.
5. Seleccione las versiones del objeto y, a continuación, elija Eliminar.
6. Para confirmar la eliminación definitiva de la versión del objeto enumerada en Objetos especificados en el cuadro de texto ¿Eliminar objetos?, introduzca Eliminar definitivamente. Amazon S3 eliminará las versiones de objetos específicos de forma permanente.

Eliminación permanente de los objetos de un bucket de Amazon S3 que no tienen habilitado el control de versiones

> **Warning**
>
> Si elimina un objeto de forma definitiva en Amazon S3, la eliminación no se puede deshacer. Además, en el caso de los buckets que no tengan el control de versiones activado, las eliminaciones son definitivas.

1. Inicie sesión AWS Management Console Management Console y abra la consola de Amazon S3 en https://console.aws.amazon.com/s3/.
2. En la lista Nombre del bucket, seleccione el nombre del bucket del que desea eliminar los objetos.
3. Seleccione los objetos y, a continuación, elija Eliminar.

4. Para confirmar la eliminación permanente de los objetos enumerados en Objetos especificados en el cuadro de texto ¿Eliminar objetos?, introduzca Eliminar permanentemente.

> **Note**
>
> Si tiene problemas para eliminar los objetos, consulte Quiero eliminar los objetos versionados de forma permanente.

Uso de los AWS SDK

Para ver ejemplos de cómo eliminar varios objetos con AWS SDK, consulte Uso de `DeleteObjects` con un AWS SDK o la CLI.

Para obtener información general sobre el uso de diferentes SDK de AWS, consulte Desarrollo con Amazon S3 mediante los SDK de AWS.

Uso de la API de REST

Puede utilizar los AWS SDK para eliminar varios objetos con la API de Multi-Object Delete. Sin embargo, si su aplicación lo requiere, puede enviar solicitudes REST directamente.

Para obtener más información, consulte Eliminar varios objetos en la referencia de API de Amazon Simple Storage Service.

Organizar, describir y trabajar con los objetos

En Amazon S3, puede utilizar prefijos para organizar el almacenamiento. Un prefijo es una agrupación lógica de los objetos en un bucket. El valor de prefijo es similar a un nombre de directorio que permite almacenar datos parecidos en el mismo directorio de un bucket. Cuando carga objetos mediante programación, puede usar prefijos para organizar los datos.

En la consola de Amazon S3, los prefijos se denominan carpetas. Puede ver todos los objetos y carpetas en la consola de S3 si se dirige al bucket. También puede ver información sobre cada objeto, incluidas las propiedades del objeto.

Para obtener más información sobre cómo describir y organizar sus datos en Amazon S3, consulte los siguientes temas.

Temas

Organizar objetos con prefijos

Puede utilizar prefijos para organizar los datos que almacena en los buckets de Amazon S3. Un prefijo es una cadena de caracteres al principio del nombre de la clave de objeto. Los prefijos pueden tener cualquier longitud, dentro de la longitud máxima permitida para los nombres de la clave de objeto (1024 bytes). Puede considerar los prefijos como una forma de organizar los datos similar a los directorios. No obstante, los prefijos no son directorios.

La búsqueda por prefijos limita los resultados solo a aquellas claves que comiencen con el prefijo especificado. El delimitador hace que una operación de lista acumule todas las claves que comparten un prefijo común en un solo resultado de lista de resumen.

El objetivo de los parámetros de prefijo y delimitador es ayudarle a organizar sus claves jerárquicamente y explorarlas. Para hacerlo, escoja un delimitador para su bucket, como una barra inclinada (/), que no esté presente en ninguno de sus nombres de clave esperados. Puede usar otro carácter como delimitador. No hay nada especial en el carácter de la barra inclinada (/), pero es un delimitador de prefijos muy común. A continuación, cree sus nombres de claves concatenando todos los niveles que contenga la jerarquía, separando cada nivel con el delimitador.

Por ejemplo, si estuviera almacenando información sobre ciudades, puede que quiera organizadla de forma natural por continente, país y provincia o estado. Dado que estos nombres no suelen contener puntuación, podría usar la barra inclinada (/) como delimitador. En el siguiente ejemplo se usa la barra inclinada (/) como delimitador:

- Europa/Francia/Nueva Aquitania/Burdeos
- Norteamérica/Canadá/Quebec/Montreal
- Norteamérica/EE. UU./Washintgon/Bellevue
- Norteamérica/EE. UU./Washintgon/Seattle

Si almacena datos para todas las ciudades del mundo siguiendo este esquema, sería extraño administrar un espacio de nombres plano para las claves. Si utiliza `Prefix` y `Delimiter` en la operación de lista, puede aprovechar la jerarquía que ha creado para enumerar los datos. Por ejemplo, para enumerar todas las zonas de EE. UU., establezca `Delimiter='/'` y `Prefix='North America/USA/'`. Para enumerar todas las provincias de Canadá para las que tenga datos, establezca `Delimiter='/'` y `Prefix='North America/Canada/'`.

Para obtener más información sobre delimitadores, prefijos y carpetas anidadas, consulte Diferencia entre los prefijos y las carpetas anidadas.

Descripción de objetos con prefijos y delimitadores

Si emite una solicitud de lista con un delimitador, puede explorar su jerarquía solo en un nivel, omitiendo y resumiendo las claves (posiblemente millones de ellas) anidadas en niveles más profundos. Por ejemplo, supongamos que tiene un bucket (*DOC-EXAMPLE-BUCKET*) con las siguientes claves:

```
sample.jpg

photos/2006/January/sample.jpg

photos/2006/February/sample2.jpg

photos/2006/February/sample3.jpg

photos/2006/February/sample4.jpg
```

El bucket de ejemplo solo tiene el objeto `sample.jpg` en el nivel raíz. Para enumerar solo los objetos en el nivel raíz del bucket, envíe una solicitud GET al bucket con el carácter delimitador barra (`/`). En respuesta, Amazon S3 devuelve la clave de objeto `sample.jpg` porque no contiene el carácter delimitador `/`. Todas las demás claves contienen el carácter delimitador. Amazon S3 agrupa estas claves y devuelve un solo elemento `CommonPrefixes` con un valor de prefijo `photos/`, que es una subcadena desde el comienzo de estas claves hasta la primera instancia del delimitador especificado.

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>
```

Para obtener más información sobre cómo describir claves de objeto mediante programación, consulte Descripción de claves de objeto mediante programación.

Descripción de claves de objeto mediante programación

En Amazon S3, las claves se pueden describir por prefijo. Puede elegir un prefijo común para los nombres de claves relacionadas y marcar estas claves con un carácter especial que delimite la jerarquía. A continuación, puede utilizar la operación de lista para seleccionar y examinar las claves jerárquicamente. La operación es similar a cómo se almacenan los archivos en directorios de un sistema de archivos.

Amazon S3 expone una operación de lista que le permite enumerar las claves que contiene un bucket. Las claves se seleccionan para la lista por bucket y prefijo. Por ejemplo, considere un bucket llamado “`dictionary`” que contiene una clave para cada palabra en inglés. Podría realizar una llamada para enumerar todas las claves en ese bucket que comiencen por la letra "q". Los resultados de listas siempre se devuelven en orden binario UTF-8.

Tanto las operaciones de lista SOAP como REST devuelven un documento XML que contiene los nombres de las claves coincidentes e información sobre el objeto identificado por cada clave.

> **Note**
>
> La compatibilidad con SOAP por HTTP está obsoleta, pero SOAP aún se encuentra disponible con HTTPS. Las características nuevas de Amazon S3 no son compatibles con SOAP. En vez de usar SOAP, le recomendamos que utilice la API de REST o los SDK de AWS.

Los grupos de claves que comparten un prefijo que termina con un delimitador especial se pueden acumular por el prefijo común para realizar las listas. Esto permite que las aplicaciones organicen y exploren sus claves jerárquicamente, del mismo modo que se organizarían los archivos en directorios en un sistema de archivos.

Por ejemplo, para ampliar el bucket dictionary de modo que contenga más palabras, además de las que están en inglés, podría formar claves si pone un prefijo a cada palabra con su idioma y un delimitador, como "`French/logical`". Con este esquema de nomenclatura y la función de listado jerárquico, podría recuperar una lista de palabras solo en francés. También podría explorar la lista de nivel superior de idiomas disponibles sin tener que iterar por todas las claves que intervienen lexicográficamente. Para obtener más información sobre este aspecto de la descripción, consulte Organizar objetos con prefijos.

API de REST

Si su aplicación lo requiere, puede enviar solicitudes REST directamente. Puede enviar una solicitud GET para devolver algunos o todos los objetos de un bucket, o puede usar criterios de selección para devolver un subconjunto de los objetos en un bucket. Para obtener más información, consulte Bucket GET (List Objects) versión 2 en la Referencia de API de Amazon Simple Storage Service.

Eficacia de implementación de listas

El rendimiento de la lista no se ve afectado sustancialmente por el número total de claves en el bucket. Tampoco se ve afectado por la presencia o ausencia de los argumentos `prefix`, `marker`, `maxkeys` o `delimiter`.

Iteración en resultados de varias páginas

Dado que los buckets pueden contener un número virtualmente ilimitado de claves, los resultados completos de una consulta de lista pueden ser extremadamente grandes. Para administrar conjuntos de resultados grandes, la API de Amazon S3 admite la paginación, de modo que los divide en varias

respuestas. Cada respuesta de lista de claves devuelve una página con hasta 1000 claves y un indicador que identifica si la respuesta está incompleta. Enviará una serie de solicitudes de listas de claves hasta que haya recibido todas las claves. AWS Las bibliotecas de encapsulamiento de los SDK facilitan la misma paginación.

Ejemplos

Los siguientes ejemplos de código muestran cómo utilizar `ListObjects`.

CLI

AWS CLI

En el siguiente ejemplo se utiliza el comando `list-objects` para mostrar los nombres de todos los objetos del bucket especificado:

```
aws s3api list-objects --bucket text-content --query 'Contents[].{Key: Key, Size: Size}'
```

En el ejemplo se utiliza el argumento `--query` para filtrar la salida de `list-objects` hasta el valor de la clave y el tamaño de cada objeto

Para obtener más información sobre los objetos, consulte Trabajo con objetos de Amazon S3 en la Guía para desarrolladores de Amazon S3.

- Para obtener detalles de la API, consulte ListObjects en la Referencia de comandos de AWS CLI.

PowerShell

Herramientas para PowerShell

Ejemplo 1: este comando recupera la información sobre todos los elementos del bucket “test-files”.

```
Get-S3Object -BucketName test-files
```

Ejemplo 2: este comando recupera la información sobre el elemento “sample.txt” del bucket “test-files”.

```
Get-S3Object -BucketName test-files -Key sample.txt
```

Ejemplo 3: este comando recupera la información sobre todos los elementos con el prefijo "sample" del bucket "test-files".

```
Get-S3Object -BucketName test-files -KeyPrefix sample
```

- Para obtener información sobre la API, consulte ListObjects en la Referencia de Cmdlet de AWS Tools for PowerShell.

Organización de objetos en la consola de Amazon S3 con carpetas

En Amazon S3, los buckets y objetos son los principales recursos, y los objetos se almacenan en buckets. Amazon S3 tiene una estructura sin formato en lugar de una jerarquía como la que vería en un sistema de archivos. Sin embargo, para la simplicidad organizativa, la consola de Amazon S3 admite el concepto de carpetas como medio para agrupar objetos. La consola lo hace utilizando un prefijo de nombre compartido para los objetos agrupados. En otras palabras, los objetos agrupados tienen nombres que comienzan por una cadena común. Esta cadena común, o prefijo compartido, es el nombre de la carpeta. Los nombres de objetos también se denominan nombres de clave.

Por ejemplo, puede crear una carpeta en la consola denominada `photos` y almacenar un objeto denominado `myphoto.jpg` en ella. El objeto luego se guarda con el nombre de clave `photos/myphoto.jpg`, donde el prefijo es `photos/`.

A continuación se incluyen dos ejemplos más:

- Si tiene tres objetos en su bucket, `logs/date1.txt`, `logs/date2.txt` y `logs/date3.txt`, la consola mostrará una carpeta con el nombre `logs`. Si abre la carpeta en la consola, verá tres objetos: `date1.txt`, `date2.txt` y `date3.txt`.
- Si tiene un objeto llamado `photos/2017/example.jpg`, la consola mostrará una carpeta denominada `photos` que contiene la carpeta `2017`. La carpeta `2017` contendrá el objeto `example.jpg`.

Puede tener carpetas dentro de carpetas, pero no buckets dentro de buckets. Puede cargar y copiar objetos directamente en una carpeta. Puede crear, eliminar y hacer públicas las carpetas, pero no les puede cambiar el nombre. Los objetos se pueden copiar de una carpeta a otra.

Important

Cuando crea una carpeta en Amazon S3, S3 crea un objeto de 0 bytes con una clave establecida en el nombre de la carpeta que ha proporcionado. Por ejemplo, si crea una carpeta denominada `photos` en el bucket, la consola de Amazon S3 crea un objeto de 0 bytes con la clave `photos/`. La consola crea este objeto para admitir la idea de carpetas. La consola de Amazon S3 trata como una carpeta a todos los objetos que tienen un carácter de barra inclinada (/) como último carácter (final) en el nombre de clave (por ejemplo `examplekeyname/`). No se puede cargar un objeto que tiene un nombre de clave con un carácter / final mediante la consola de Amazon S3. Sin embargo, los objetos cuyos nombres incluyen una / final se pueden cargar con la API de Amazon S3 a través de la AWS Command Line Interface (AWS CLI), los SDK de AWS o la API de REST.
Un objeto cuyo nombre incluye una / final se muestra como una carpeta en la consola de Amazon S3. La consola de Amazon S3 no muestra el contenido ni los metadatos para dicho objeto. Si se usa la consola para copiar un objeto cuyo nombre incluye una / final, se crea una nueva carpeta en la ubicación de destino pero los datos y metadatos del objeto no se copian.

Temas

Creación de una carpeta

En esta sección se describe cómo utilizar la consola de Amazon S3 para crear una carpeta.

Important

Si la política de buckets impide cargar objetos en este bucket sin etiquetas, metadatos ni listas de control de acceso (ACL), no podrá crear una carpeta mediante el procedimiento siguiente. En lugar de eso, cargue una carpeta vacía y especifique las siguientes opciones en la configuración de carga.

Para crear una carpeta

1. Inicie sesión AWS Management Console Management Console y abra la consola de Amazon S3 en https://console.aws.amazon.com/s3/.
2. En el panel de navegación izquierdo, elija Instancias.
3. En la lista Buckets, seleccione el nombre del bucket donde desea crear una carpeta.
4. Si su política de bucket impide cargar objetos a este bucket sin cifrado, debe elegir Enable (Habilitar) en Server-side encryption (Cifrado del lado del servidor).
5. Elija Crear carpeta.
6. Escriba un nombre para la carpeta (por ejemplo, **`favorite-pics`**). Luego, elija Create folder (Crear carpeta).

Hacer públicas las carpetas

Le recomendamos bloquear todo el acceso público a sus carpetas de Amazon S3 y buckets a menos que requiera específicamente una carpeta o bucket público. Al hacer pública una carpeta, cualquier persona en Internet puede ver todos los objetos que están agrupados en dicha carpeta.

En la consola de Amazon S3, puede hacer pública una carpeta. También puede hacer pública una carpeta creando una política de bucket que limite el acceso a los datos mediante prefijo. Para obtener más información, consulte Administración de identidades y accesos para Amazon S3.

> **Warning**
>
> Después de hacer una carpeta pública en la consola de Amazon S3, no puede volver a hacerla privada. En lugar de ello, debe definir permisos en cada objeto individual en la carpeta pública para que los objetos no tengan acceso público. Para obtener más información, consulte Configuración de la ACL.

Temas

- Calcular tamaño de carpeta
- Eliminación de carpetas

Calcular tamaño de carpeta

En esta sección se describe cómo utilizar la consola de Amazon S3 para calcular el tamaño de una carpeta.

Para calcular el tamaño de una carpeta

1. Inicie sesión AWS Management Console Management Console y abra la consola de Amazon S3 en https://console.aws.amazon.com/s3/.
2. En el panel de navegación izquierdo, elija Instancias.
3. En la lista Buckets, elija el nombre del bucket en el que se almacena la carpeta.
4. En la lista Objects (Objetos), seleccione la casilla de verificación junto al nombre de la carpeta.
5. Elija Actions (Acciones) y, a continuación, elija Calculate total size (Calcular el tamaño total).

> **Note**
>
> Cuando salga de la página, la información de la carpeta (incluido el tamaño total) dejará de estar disponible. Deberá calcular de nuevo el tamaño total si quiere volver a verlo.

> **Important**
>
> Cuando utiliza la acción Calculate total size (Calcular tamaño total) en objetos o carpetas específicos de su bucket, Amazon S3 calcula el número total de objetos y el tamaño total de almacenamiento. Sin embargo, las cargas multiparte incompletas o en curso y las versiones anteriores o no actuales no se calculan en función del número total de objetos ni del tamaño total. Esta acción calcula solo el número total de objetos y el tamaño total de la versión actual o más reciente de cada objeto almacenado en el bucket.
> Por ejemplo, si hay dos versiones de un objeto en su bucket, la calculadora de almacenamiento de Amazon S3 las cuenta como un solo objeto. Como resultado, el número total de objetos que se calcula en la consola de Amazon S3 puede diferir de la métrica de Object Count (Recuento de objetos) que se muestra en S3 Storage Lens y del número reportado por la métrica de Amazon CloudWatch, `NumberOfObjects`. Del mismo modo, el tamaño total del almacenamiento también puede diferir de la métrica de Total Storage (Almacenamiento total) que se muestra en S3 Storage Lens y de la métrica de `BucketSizeBytes` que se muestra en CloudWatch.

Eliminación de carpetas

En esta sección se explica cómo utilizar la consola de Amazon S3 para eliminar carpetas de un bucket de S3.

Para obtener información sobre las características y precios de Amazon S3, consulte Amazon S3.

Para eliminar carpetas de un bucket de S3

1. Inicie sesión AWS Management Console Management Console y abra la consola de Amazon S3 en https://console.aws.amazon.com/s3/.
2. En la lista Buckets, seleccione el nombre del bucket cuyas carpetas desea eliminar.
3. En la lista Objects (Objetos), active la casilla de verificación situada junto a las carpetas y objetos que desea eliminar.
4. Elija Delete (Eliminar).
5. En la página Delete objects (Eliminar objetos), compruebe que aparezcan los nombres de las carpetas que seleccionó para eliminar.
6. En el cuadro Eliminar objetos, escriba **`delete`** y elija Eliminar objetos.

> **Warning**
>
> Esta acción elimina todos los objetos especificados. Al eliminar carpetas, espere a que finalice la acción de eliminación antes de agregar nuevos objetos a la carpeta. De lo contrario, es posible que también se eliminen objetos nuevos.

Visualización de información general sobre objetos en la consola de Amazon S3

Puede utilizar la consola de Amazon S3 para ver información general sobre un objeto. La consola proporciona toda la información esencial de un objeto en un solo lugar.

Para abrir la página de detalles de un objeto

1. Inicie sesión en la AWS Management Console y abra la consola de Amazon S3 en https://console.aws.amazon.com/s3/.

2. En la lista Buckets (Buckets), elija el nombre del bucket que contiene el objeto.
3. En la lista Objects (Objetos), seleccione el nombre del objeto para el que desea obtener información general.

   Se abrirá la página de detalles del objeto.
4. Para descargar el objeto, elija Object actions (Acciones de objeto) y, a continuación, elija Download (Descargar). Para copiar la ruta del objeto en el portapapeles, en Object URL (URL de objeto), elija la URL.
5. Si el control de versiones está habilitado en el bucket, elija Versions (Versiones) para mostrar las versiones del objeto.

   - Para descargar una versión de objeto, active la casilla situada junto al ID de versión, elija Actions (Acciones) y, a continuación, elija Download (Descargar).
   - Para eliminar una versión de objeto, active la casilla de verificación situada junto al ID de versión y elija Delete (Eliminar).

   > **Important**
   >
   > Solo puede anular la eliminación de un objeto si se ha eliminado en su última versión (la más reciente). No puede anular la eliminación de una versión anterior de un objeto que se haya eliminado.

Visualización de propiedades de objeto en la consola de Amazon S3

Puede utilizar la consola de Amazon S3 para ver las propiedades de un objeto, incluida la clase de almacenamiento, la configuración de cifrado, las etiquetas y los metadatos.

Para ver las propiedades de un objeto:

1. Inicie sesión en la AWS Management Console y abra la consola de Amazon S3 en https://console.aws.amazon.com/s3/.
2. En la lista Buckets (Buckets), elija el nombre del bucket que contiene el objeto.
3. En la lista Objects (Objetos), elija el nombre del objeto para el que desea ver las propiedades.

   Se abrirá la Object overview (Información general sobre el objeto) para el objeto. Puede desplazarse hacia abajo para ver las propiedades del objeto.

4. En la página Object overview (Información general del objeto), puede configurar las siguientes propiedades para el objeto.

> **Note**
>
> - Si cambia las propiedades de clase de almacenamiento, cifradoo metadatos, se crea un nuevo objeto para reemplazar el antiguo. Si el control de versiones de S3 está activado, se crea una nueva versión del objeto y el objeto existente se convierte en una versión anterior. El rol que cambia la propiedad también se convierte en el propietario del nuevo objeto o (versión del objeto).
> - Si cambia las propiedades de Clase de almacenamiento, Cifrado o Metadatos de un objeto que tiene etiquetas definidas por el usuario, debe tener el permiso `s3:GetObjectTagging`. Si va a cambiar estas propiedades de un objeto que no tiene etiquetas definidas por el usuario pero que tiene un tamaño superior a 16 MB, también debe tener el permiso `s3:GetObjectTagging`.
>
>   Si la política de bucket de destino deniega la acción `s3:GetObjectTagging`, estas propiedades del objeto se actualizarán, pero las etiquetas definidas por el usuario se eliminarán del objeto y aparecerá un error.

a. Storage class (Clase de almacenamiento): todos los objetos de Amazon S3 tienen una clase de almacenamiento asociada. La clase de almacenamiento que quiera usar dependerá de la frecuencia con la que obtenga acceso al objeto. La clase predeterminada de almacenamiento para objetos de S3 es STANDARD. Puede seleccionar qué clase de almacenamiento usar al cargar un objeto. Para obtener más información acerca de las clases de almacenamiento, consulte Uso de las clases de almacenamiento de Amazon S3.

   Para cambiar la clase de almacenamiento tras cargar un objeto, seleccione Storage class (Clase de almacenamiento). Seleccione la clase de almacenamiento que desee y haga clic en Save (Guardar).

b. Configuración de cifrado del lado del servidor: puede utilizar el cifrado del lado del servidor para cifrar los objetos de S3. Para obtener más información, consulte Especificación del cifrado del lado del servidor con AWS KMS (SSE-KMS) o Especificación del cifrado del servidor con claves administradas por Amazon S3 (SSE-S3).

c. Metadata (Metadatos): cada objeto de Amazon S3 tiene un conjunto de pares nombre-valor que representan sus metadatos. Para obtener información sobre cómo agregar metadatos a un objeto de S3, consulte Edición de metadatos de objeto en la consola de Amazon S3.

d. Etiquetas: puede clasificar el almacenamiento si agrega etiquetas a un objeto de S3. Para obtener más información, consulte Categorización del almacenamiento mediante etiquetas.

e. Retención y retención legal de bloqueo de objetos: puede evitar que se elimine un objeto. Para obtener más información, consulte Usar Bloqueo de objetos de S3.

Uso de URL prefirmadas

Puede utilizar direcciones URL prefirmadas para otorgar acceso limitado en el tiempo a objetos en Amazon S3 sin necesidad de actualizar su política de buckets. Una URL prefirmada puede introducirse en un navegador o utilizarse por un programa para descargar un objeto. Las credenciales que utiliza la URL prefirmada son las del usuario de AWS que generó la URL.

También puede utilizar URL prefirmadas para permitir que alguien cargue un objeto específico en su bucket de Amazon S3. Esto permite una carga sin necesidad de que otra parte tenga credenciales o permisos de seguridad de AWS. Si ya existe un objeto con la misma clave en el bucket especificado en la URL prefirmada, Amazon S3 reemplazará el objeto existente por el objeto cargado.

Puede utilizar la URL prefirmada varias veces, hasta la fecha y hora de vencimiento.

Cuando crea una URL prefirmada, debe proporcionar sus credenciales de seguridad y luego especificar lo siguiente:

- Un bucket de Amazon S3.
- Una clave de objeto (si está descargando, este objeto estará en su bucket de Amazon S3; si está cargando, este es el nombre del archivo que se cargará).
- Un método HTTP (`GET` para descargar objetos o `PUT` para cargarlos)
- Un intervalo de tiempo de vencimiento

Actualmente, las URL prefirmadas de Amazon S3 no admiten el uso de los siguientes algoritmos de suma de comprobación de la integridad de los datos (CRC32, CRC32C, SHA-1, SHA-256) al cargar objetos. Para verificar la integridad del objeto después de cargarlo, puede proporcionar un resumen MD5 del objeto al cargarlo con una URL prefirmada. Para obtener más información acerca de la integración de objetos, consulte Comprobación de la integridad de objetos.

Temas

Quién puede crear una URL prefirmada

Cualquiera que tenga credenciales de seguridad válidas puede crear una URL prefirmada. Sin embargo, para poder obtener acceso a un objeto correctamente, la URL prefirmada debe haber sido creada por alguien que tenga permiso para realizar la operación en la que se basa la URL prefirmada.

Los siguientes son los tipos de credenciales que puede utilizar para crear una URL prefirmada:

- Perfil de instancia de IAM: válido hasta 6 horas.
- AWS Security Token Service: válido hasta un máximo de 36 horas cuando se firma con credenciales de seguridad de larga duración o la duración de la credencial temporal, lo que termine antes.
- Usuario de IAM: válido hasta 7 días cuando se utiliza AWS Signature Version 4.

  Para crear una URL prefirmada válida durante un máximo de 7 días, delegue primero las credenciales de usuario de IAM (la clave de acceso y la clave secreta) en el método que va a utilizar para crear la URL prefirmada.

> **Note**
>
> Si creó una URL prefirmada con una credencial temporal, la URL caducará cuando caduque la credencial. En general, una URL prefirmada caduca cuando se revoca, elimina o desactiva la credencial que utilizó para crearla. Esto ocurre incluso si la URL se creó con un tiempo de caducidad posterior. Para conocer la duración temporal de las credenciales de seguridad, consulte Comparación de las operaciones de la API de AWS STS en la Guía del usuario de IAM.

Tiempo de caducidad de las URL prefirmadas

Una URL prefirmada sigue siendo válida durante el período de tiempo especificado cuando se generó la URL. Si crea una URL prefirmada con la consola de Amazon S3, el tiempo de caducidad puede establecerse entre 1 minuto y 12 horas. Si utiliza la AWS CLI o los SDK de AWS, el tiempo de caducidad puede establecerse hasta en 7 días.

Si creó una URL prefirmada con un token temporal, la URL caducará cuando caduque el token. En general, una URL prefirmada caduca cuando se revoca, elimina o desactiva la credencial que utilizó para crearla. Esto ocurre incluso si la URL se creó con un tiempo de caducidad posterior. Para obtener más información sobre cómo las credenciales que utiliza afectan al tiempo de caducidad, consulte Quién puede crear una URL prefirmada.

Simple Storage Service (Amazon S3) comprueba la fecha y hora de vencimiento de una URL firmada al realizarse la solicitud HTTP. Por ejemplo, si un cliente comienza a descargar un archivo grande inmediatamente antes de la fecha de vencimiento, la descarga continúa incluso si se sobrepasa la hora de vencimiento durante la descarga. Sin embargo, si la conexión se interrumpe y el cliente intenta reiniciar la descarga después de la hora de vencimiento, la descarga produce un error.

Limitación de las capacidades de URL prefirmadas

Las capacidades de una URL están limitadas por los permisos del usuario que la creó. En esencia, las URL prefirmadas son tokens al portador que otorgan acceso a quienes las poseen. Por lo tanto, le recomendamos que los proteja adecuadamente. A continuación, se muestran algunos métodos que puede utilizar para restringir el uso de las URL prefirmadas.

AWS Signature Version 4 (SigV4)

Para aplicar un comportamiento específico cuando las solicitudes de URL prefirmadas se autentican mediante AWS Signature Version 4 (SigV4), puede usar claves de condición en las políticas de bucket y en las políticas de punto de acceso. Por ejemplo, puede crear la siguiente política de buckets que use la condición `s3:signatureAge` para denegar cualquier solicitud de URL prefirmada de Amazon S3 en los objetos del bucket *`amzn-s3-demo-bucket1`* si la firma tiene más de 10 minutos de antigüedad. Para utilizar este ejemplo, reemplace los *`user input placeholders`* con su propia información.

```
{
   "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:::amzn-s3-demo-bucket1/*",
        "Condition": {
           "NumericGreaterThan": {
              "s3:signatureAge": 600000
           }
        }
     }
   ]
}
```

Para obtener más información sobre las claves de política relacionadas con AWS Signature Version 4, consulte Autenticación de AWS Signature Version 4 en la Referencia de la API de Amazon Simple Storage Service.

Restricción de ruta de red

Si desea restringir el uso de URL prefirmadas y todo el acceso a Amazon S3 a rutas de red concretas, puede escribir políticas de AWS Identity and Access Management (IAM). Estas políticas se pueden establecer en la entidad principal de IAM que realiza la llamada, en el bucket de Amazon S3 o en ambos.

Una restricción de ruta de red en la entidad principal de IAM requiere que el usuario de esas credenciales realice solicitudes desde la red especificada. Una restricción en el bucket o en el punto de acceso requiere que todas las solicitudes a ese recurso se originen desde la red especificada. Estas restricciones también se aplican fuera del escenario de URL prefirmada.

La clave de condición global de IAM que utilice depende del tipo de punto de conexión. Si utiliza el punto de conexión público para Amazon S3, utilice `aws:SourceIp`. Si utiliza un punto de conexión de la nube privada virtual (VPC) para Amazon S3, utilice `aws:SourceVpc` o `aws:SourceVpce`.

La siguiente instrucción de política de IAM requiere que la entidad principal acceda a AWS solo desde el rango de red especificado. Con esta declaración de política, todo acceso debe originarse desde ese rango. Esto incluye el caso de alguien que usa una URL prefirmada para Amazon S3. Para utilizar este ejemplo, reemplace los *`user input placeholders`* con su propia información.

```
{
```

```
    "Sid": "NetworkRestrictionForIAMPrincipal",
    "Effect": "Deny",
    "Action": "*",
    "Resource": "*",
    "Condition": {
        "NotIpAddressIfExists": {"aws:SourceIp": "IP-address-range"},
        "BoolIfExists": {"aws:ViaAWSService": "false"}
    }
}
```

Uso compartido de objetos con URL prefirmadas

De forma predeterminada, todos los objetos de Amazon S3 son privados y solo el propietario del objeto tiene permiso para obtener acceso a ellos. No obstante, el propietario del objeto puede compartirlo con otros creando una URL prefirmada. Una URL prefirmada utiliza credenciales de seguridad para conceder permisos limitados en el tiempo para descargar objetos. La URL puede introducirse en un navegador o puede utilizarla un programa para descargar el objeto. Las credenciales que utiliza la URL prefirmada son las del usuario de AWS que generó la URL.

Para obtener información general sobre las URL prefirmadas, consulte Uso de URL prefirmadas.

Puede crear una URL prefirmada para compartir un objeto sin escribir código mediante la consola de Amazon S3, AWS Explorer para Visual Studio (Windows) o AWS Toolkit for Visual Studio Code. También puede generar una URL prefirmada mediante programación con la AWS Command Line Interface (AWS CLI) o los SDK de AWS.

Uso de la consola de S3

Puede utilizar la consola de Amazon S3 para generar una URL prefirmada para compartir un objeto si sigue estos pasos. Al utilizar la consola, el tiempo máximo de caducidad de una URL prefirmada es de 12 horas desde el momento de su creación.

Para generar una URL prefirmada con la consola de Amazon S3

1. Inicie sesión AWS Management Console Management Console y abra la consola de Amazon S3 en https://console.aws.amazon.com/s3/.
2. En el panel de navegación izquierdo, elija Instancias.
3. En la lista Buckets (Buckets), elija el nombre del bucket que contiene el objeto para el que desea generar una URL prefirmada.
4. En la lista Objects (Objetos), seleccione el objeto para el que desea crear una URL prefirmada.

5. En el menú Acciones, elija Compartir con una URL prefirmada.
6. Especifique cuánto tiempo de validez desea que tenga la URL prefirmada.
7. Elija Create presigned URL (Crear URL prefirmada).
8. Cuando aparece una confirmación, la URL se copia automáticamente en el portapapeles. Verá un botón para copiar la URL prefirmada en caso de que necesite volver a copiarla.

Uso de la AWS CLI

El siguiente comando de ejemplo de la AWS CLI genera una URL prefirmada para compartir un objeto de un bucket de Amazon S3. Cuando se utiliza la AWS CLI, el tiempo máximo de vencimiento de una URL prefirmada es de 7 días desde el momento de su creación. Para utilizar este ejemplo, reemplace los *user input placeholders* con su propia información.

```
aws s3 presign s3://amzn-s3-demo-bucket1/mydoc.txt --expires-in 604800
```

> **Note**
>
> Para todas las Regiones de AWS lanzadas después del 20 de marzo de 2019, es necesario especificar `endpoint-url` y `Región de AWS` con la solicitud. Para obtener una lista de todas las regiones y puntos de conexión de Amazon S3, consulte Regiones y puntos de conexión en la Referencia general de AWS.

```
aws s3 presign s3://amzn-s3-demo-bucket1/mydoc.txt --expires-in 604800 --region af-south-1 --endpoint-url https://s3.af-south-1.amazonaws.com
```

Para obtener más información, consulte presign en la Referencia de los comandos de AWS CLI.

Uso de los AWS SDK

Para ver ejemplos de cómo utilizar los SDK de AWS para generar una URL prefirmada para compartir un objeto, consulte Crear una URL prefirmada para Amazon S3 mediante un SDK de AWS.

Cuando utilice los SDK de AWS para generar una URL prefirmada, el tiempo máximo de caducidad es de 7 días desde el momento de su creación.

> **Note**
>
> Para todas las Regiones de AWS lanzadas después del 20 de marzo de 2019, es necesario especificar `endpoint-url` y `Región de AWS` con la solicitud. Para obtener una lista de todas las regiones y puntos de conexión de Amazon S3, consulte Regiones y puntos de conexión en la Referencia general de AWS.

> **Note**
>
> Al utilizar los SDK de AWS, el atributo de etiquetado debe ser un encabezado y no un parámetro de consulta. Todos los demás atributos se pueden pasar como parámetros para la URL prefirmada.

Uso de AWS Toolkit for Visual Studio (Windows)

> **Note**
>
> En este momento, AWS Toolkit for Visual Studio no es compatible con Visual Studio para Mac.

1. Instale AWS Toolkit for Visual Studio siguiendo las instrucciones de Installing and setting up the Toolkit for Visual Studio en la Guía del usuario de AWS Toolkit for Visual Studio.
2. Conéctese a AWS con los pasos siguientes, Conexión a AWS en la Guía del usuario de AWS Toolkit for Visual Studio.
3. En el panel lateral izquierdo que tiene la etiqueta Explorador de AWS, haga doble clic en el bucket que contiene el objeto.
4. Haga clic con el botón derecho en el objeto para el que desee que se genere una URL prefirmada y seleccione Crear URL prefirmada...
5. En la ventana emergente, defina la fecha y la hora de vencimiento de su URL prefirmada.
6. La Clave de objeto debería rellenarse previamente en función del objeto que haya seleccionado.
7. Elija GET para especificar que esta URL prefirmada se utilizará para descargar un objeto.
8. Elija el botón Generate (Generar).

9. Para copiar la URL en el portapapeles, elija Copy (Copiar).
10. Para usar la URL prefirmada generada, pegue la URL en cualquier navegador.

Uso de AWS Toolkit for Visual Studio Code

Si utiliza Visual Studio Code, puede generar una URL prefirmada para compartir un objeto sin escribir código mediante AWS Toolkit for Visual Studio Code. Para obtener información general, consulte AWS Toolkit for Visual Studio Code en la Guía del usuario de AWS Toolkit for Visual Studio Code.

Para obtener instrucciones acerca de cómo instalar AWS Toolkit for Visual Studio Code, consulte Instalación de AWS Toolkit for Visual Studio Code en la Guía del usuario de AWS Toolkit for Visual Studio Code.

1. Conéctese a AWS con los pasos siguientes, Conexión a AWS Toolkit for Visual Studio Code en la Guía del usuario de AWS Toolkit for Visual Studio Code.
2. Seleccione el logotipo de AWS en el panel izquierdo en Visual Studio Code.
3. En EXPLORADOR, seleccione S3.
4. Elija un bucket y un archivo y abra el menú contextual (botón derecho del ratón).
5. Elija Generar URL prefirmada y, a continuación, establezca el tiempo de caducidad (en minutos).
6. Pulse Intro y la URL prefirmada se copiará en el portapapeles.

Carga de objetos con URL prefirmadas

Puede utilizar URL prefirmadas para permitir que alguien cargue un objeto en su bucket de Amazon S3. El uso de una URL prefirmada permitirá cargar datos sin necesidad de que otra parte disponga de credenciales ni permisos de seguridad de AWS. Una URL prefirmada está limitada por los permisos del usuario que la crea. Es decir, si recibe una URL prefirmada para cargar un objeto, podrá cargarlo solamente si el creador de la URL cuenta con los permisos necesarios para ello.

Cuando alguien usa la URL para cargar un objeto, Amazon S3 crea el objeto en el bucket especificado. Si ya existe en el bucket un objeto con la misma clave que se ha especificado en la URL prefirmada, Amazon S3 reemplaza el objeto existente con el objeto cargado. Tras la carga, el propietario del bucket será el propietario del objeto.

Para obtener información general sobre las URL prefirmadas, consulte Uso de URL prefirmadas.

Puede crear URL prefirmada para cargar un objeto sin necesidad de escribir código alguno gracias a AWS Explorer para Visual Studio. Puede generar una URL prefirmada mediante programación con los SDK de AWS.

Uso de AWS Toolkit for Visual Studio (Windows)

> **Note**
>
> En este momento, AWS Toolkit for Visual Studio no es compatible con Visual Studio para Mac.

1. Instale AWS Toolkit for Visual Studio siguiendo las instrucciones de Installing and setting up the Toolkit for Visual Studio en la Guía del usuario de AWS Toolkit for Visual Studio.
2. Conéctese a AWS con los pasos siguientes, Conexión a AWS en la Guía del usuario de AWS Toolkit for Visual Studio.
3. En el panel lateral izquierdo que tiene la etiqueta Explorador de AWS, haga clic con el botón derecho en el bucket en el que desee cargar un objeto.
4. Elija Crear URL prefirmada…
5. En la ventana emergente, defina la fecha y la hora de vencimiento de su URL prefirmada.
6. En Clave de objeto, establezca el nombre del archivo que se va a cargar. El archivo que vaya a cargar debe coincidir exactamente con este nombre. Si ya existe un objeto con la misma clave de objeto en el bucket, Amazon S3 sustituirá el objeto existente por el objeto recién cargado.
7. Elija PUT para especificar que esta URL prefirmada se utilizará para cargar un objeto.
8. Elija el botón Generate (Generar).
9. Para copiar la URL en el portapapeles, elija Copy (Copiar).
10. Para usar esta URL, puede enviar una solicitud PUT con el comando `curl`. Incluya la ruta completa al archivo y la propia URL prefirmada.

```
curl -X PUT -T "/path/to/file" "presigned URL"
```

Uso de los AWS SDK

Para ver ejemplos de cómo utilizar los SDK de AWS para generar una URL prefirmada para cargar un objeto, consulte Crear una URL prefirmada para Amazon S3 mediante un SDK de AWS.

Cuando utilice los SDK de AWS para generar una URL prefirmada, el tiempo máximo de caducidad es de 7 días desde el momento de su creación.

> **Note**
>
> Para todas las Regiones de AWS lanzadas después del 20 de marzo de 2019, es necesario especificar `endpoint-url` y `Región de AWS` con la solicitud. Para obtener una lista de todas las regiones y puntos de conexión de Amazon S3, consulte Regiones y puntos de conexión en la Referencia general de AWS.

Transformación de objetos con Lambda para objetos S3

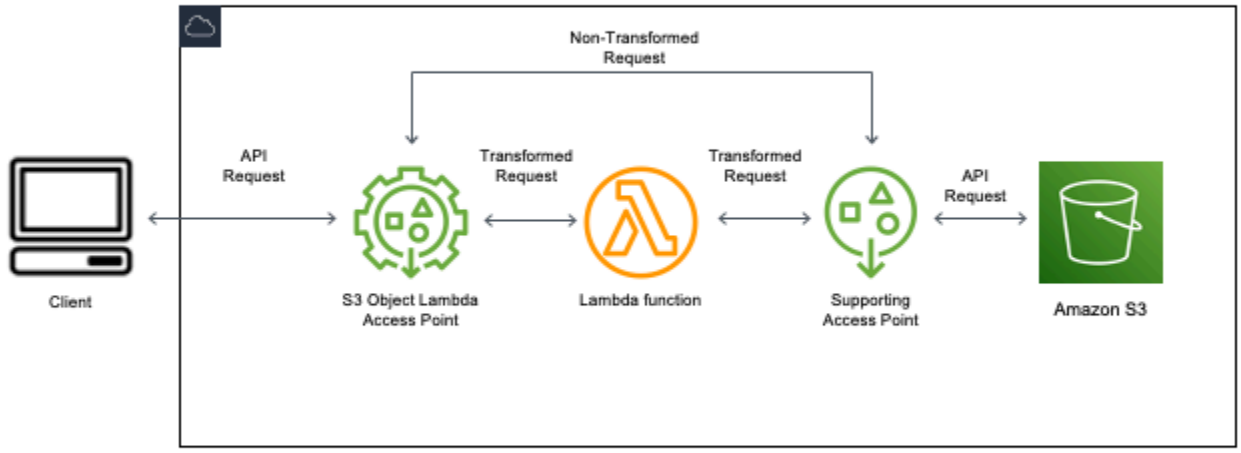
Con Amazon S3 Object Lambda, puede agregar su propio código a las solicitudes `GET`, `LIST` y `HEAD` de Amazon S3 para modificar y procesar los datos a medida que vuelven a una aplicación. Puede utilizar código personalizado para modificar los datos que devuelven solicitudes `GET` de S3 para filtrar columnas, redimensionar imágenes de forma dinámica y aplicarles marcas de agua, redactar datos confidenciales y más. También puede usar S3 Object Lambda para modificar la salida de las solicitudes `LIST` de S3 para crear una vista personalizada de todos los objetos de un bucket y las solicitudes `HEAD` de S3 para modificar los metadatos de los objetos, como el nombre y el tamaño del objeto. Puede utilizar S3 Object Lambda como origen de su distribución de Amazon CloudFront para adaptar los datos a los usuarios finales, por ejemplo, redimensionar automáticamente las imágenes, transcodificar formatos antiguos (como de JPEG a WebP) o eliminar los metadatos. Para obtener más información, consulte la publicación del blog de AWS sobre el uso de Amazon S3 Object Lambda con Amazon CloudFront. Gracias a las funciones de Lambda de AWS, su código se ejecuta en una infraestructura totalmente gestionada por AWS. El uso de S3 Object Lambda reduce la necesidad de crear y almacenar copias derivativas de sus datos o ejecutar proxy, además de no tener que cambiar sus aplicaciones.

Cómo funciona S3 Object Lambda

S3 Object Lambda utiliza funciones de AWS Lambda que permiten procesar la salida de una solicitud `GET`, `LIST` o `HEAD` estándar de S3 de forma automática. AWS Lambda se trata de un servicio informático sin servidor que ejecuta código definido por el cliente sin necesidad de administrar los recursos informáticos subyacentes. Puede crear y ejecutar sus propias funciones de Lambda personalizadas, de modo que es posible personalizar la transformación de datos a su caso de uso específico.

Después de configurar una función de Lambda, la asocia a un punto de conexión de servicio S3 Object Lambda, denominado punto de acceso del objeto Lambda. El punto de acceso del objeto Lambda utiliza un punto de acceso S3 estándar, denominado punto de acceso de apoyo, para acceder a Amazon S3.

Cuando envía una solicitud a su punto de acceso del objeto Lambda, Amazon S3 llama automáticamente a su función de Lambda. Cualquier dato obtenido con una solicitud `GET`, `LIST` o `HEAD` de S3 a través del punto de acceso del objeto Lambda devuelve un resultado transformado a la aplicación. Todas las demás solicitudes se procesan con normalidad, como se ilustra en el siguiente diagrama.



Los temas de esta sección describen cómo trabajar con S3 Object Lambda

Temas

- Creación de puntos de acceso Object Lambda
- Uso de puntos de acceso de Amazon S3 Object Lambda
- Consideraciones de seguridad para los puntos de acceso de S3 Object Lambda
- Escritura de funciones de Lambda para puntos de acceso de S3 Object Lambda
- Uso de funciones de Lambda creadas por AWS

- Prácticas recomendadas y directrices para S3 Object Lambda
- Tutoriales de S3 Object Lambda
- Depuración de S3 Object Lambda

Creación de puntos de acceso Object Lambda

Un punto de acceso de Object Lambda está asociado exactamente con un punto de acceso estándar y, por lo tanto, con un bucket de Amazon S3. Para crear un punto de acceso de Object Lambda, necesita los siguientes recursos:

- Un bucket de Amazon S3. Para obtener más información acerca de cómo se crean los buckets, consulte the section called “Crear un bucket”.
- Un punto de acceso estándar S3. Cuando se trabaja con puntos de acceso Object Lambda, este punto de acceso estándar se conoce como punto de acceso de apoyo. Para obtener información sobre la creación de puntos de acceso estándar, consulte the section called “Crear puntos de acceso”.
- Una función de AWS Lambda Puede crear su propia función de Lambda o puede utilizar una función precreada. Para obtener más información sobre la creación de funciones de Lambda, consulte the section called “Escritura de funciones de Lambda”. Para obtener más información sobre funciones prediseñadas, consulte Uso de funciones de Lambda creadas por AWS.
- (Opcional) Una política de AWS Identity and Access Management (IAM). Los puntos de acceso de Amazon S3 admiten políticas de recursos de IAM que le permiten controlar el uso del punto de acceso en función del recurso, del usuario o de otras condiciones. Para obtener más información sobre la creación de estas políticas, consulte the section called “Configuración de políticas de IAM”.

En las siguientes secciones se describe el modo de crear un punto de acceso de Object Lambda mediante:

- Con la AWS Management Console
- La AWS Command Line Interface (AWS CLI)
- Una plantilla de AWS CloudFormation
- Con la AWS Cloud Development Kit (AWS CDK)

Si quiere obtener más información para crear un punto de acceso de Object Lambda mediante la API de REST, consulte `CreateAccessPointForObjectLambda` en la referencia de la API de Amazon Simple Storage Service.

Crear un punto de acceso de Object Lambda

Utilice uno de los siguientes procedimientos para crear un punto de acceso de Object Lambda.

Uso de la consola de S3

Para crear un punto de acceso de Object Lambda mediante la consola

1. Inicie sesión AWS Management Console Management Console y abra la consola de Amazon S3 en https://console.aws.amazon.com/s3/.
2. En la barra de navegación, elija el nombre de la Región de AWS que aparece. A continuación, elija la región a la que desea cambiar.
3. En el panel de navegación del lado izquierdo de la consola, elija Object Lambda Access Points (Puntos de acceso de Object Lambda).
4. En la página Object Lambda Access Points (Puntos de acceso Object Lambda), elija Create Object Lambda Access Point (Crear un punto de acceso de Object Lambda).
5. En Object Lambda access point name (Nombre del punto de acceso de Object Lambda), introduzca el nombre que desea utilizar para el punto de acceso.

   Al igual que con los puntos de acceso estándar, existen reglas para los nombres de los puntos de acceso Object Lambda. Para obtener más información, consulte Reglas para asignar nombres a los puntos de acceso de Amazon S3.
6. En Supporting access point(Punto de acceso compatible), introduzca o busque el punto de acceso estándar que desea utilizar. El punto de acceso debe estar en la misma Región de AWS que los objetos que desea transformar. Para obtener información sobre la creación de puntos de acceso estándar, consulte the section called “Crear puntos de acceso”.
7. En Configuración de la transformación, puede agregar una función que transforme los datos de su punto de acceso de Object Lambda. Realice una de las siguientes acciones siguientes:

   - Si ya tiene una función AWS Lambda en su cuenta, puede seleccionarla en Invoke Lambda function (Invocar función de Lambda). Aquí puede introducir el nombre de recurso de Amazon (ARN) de una función de Lambda en la Cuenta de AWS o elegir una función de Lambda del menú desplegable.

- Si desea utilizar una función AWS creada, elija el nombre de la función en AWS built function (función generada por) y seleccione Create Lambda function (Crear función de Lambda). Esto lo llevará a la consola de Lambda, donde podrá implementar una función integrada en su Cuenta de AWS. Para obtener más información sobre funciones integrada, consulte Uso de funciones de Lambda creadas por AWS.

En S3 APIs (API de S3), elija una o más operaciones de API para invocar. Por cada API seleccionada, debe especificar una función de Lambda para invocarla.

8. (Opcional) En Payload (Carga), agregue el texto JSON que desea proporcionar a su función de Lambda como entrada. Puede configurar cargas con diferentes parámetros para diferentes puntos de acceso Object Lambda que invoquen la misma función de Lambda, ampliando así la flexibilidad de la función de Lambda.

> **Important**
>
> Cuando utilice los puntos de acceso Object Lambda, asegúrese de que la carga no contenga información confidencial.

9. (Opcional) Para Range and part number (Rango y número de pieza), debe habilitar esta opción si quiere procesar solicitudes `GET` y `HEAD` con encabezados de rango y número de pieza. Cuando se selecciona esta opción, se confirma que la función de Lambda puede reconocer y procesar estas solicitudes. Para obtener más información sobre los encabezados de rango y los números de pieza, consulte Trabajar con encabezados Range y partNumber.
10. (Opcional) En Métricas de solicitudes, seleccione Habilitar o Deshabilitar para agregar la supervisión de Amazon S3 al punto de acceso de Object Lambda. Las métricas de solicitud se facturan según la tarifa de Amazon CloudWatch estándar.
11. (Opcional) En Object Lambda Access Point policy (Política de punto de acceso de Object Lambda), establezca una política de recursos. Las políticas de recursos otorgan permisos para el punto de acceso de Object Lambda especificado y pueden controlar el uso del punto de acceso en función del recurso, del usuario o de otras condiciones. Para obtener más información sobre las políticas de recursos de punto de acceso Object Lambda, consulte Configuración de las políticas de IAM para puntos de acceso de Object Lambda.
12. En Block Public Access settings for this Object Lambda Access Point (Configuración de bloqueo del acceso público a este punto de acceso de Object Lambda), seleccione la configuración de bloqueo de acceso público que desee aplicar. Todas las configuraciones de bloqueo de acceso público están habilitadas de forma predeterminada para los nuevos puntos de acceso Object

Lambda, y le recomendamos dejar habilitada la configuración predeterminada. Amazon S3 actualmente no admite cambiar la configuración de bloqueo de acceso público de un punto de acceso de Object Lambda después de que se haya creado el punto de acceso de Object Lambda.

Para obtener más información sobre el uso del bloqueo de acceso público de Amazon S3, consulte Administrar el acceso público a los puntos de acceso.

13. Elija Create Object Lambda Access Point (Crear punto de acceso de Object Lambda).

Uso de la AWS CLI

Para crear un punto de acceso de Object Lambda mediante la plantilla AWS CloudFormation

> **Note**
>
> Para utilizar los comandos siguientes, sustituya *`user input placeholders`* con su información.

1. Descargue el paquete de implementación de la función de AWS Lambda `s3objectlambda_deployment_package.zip` en la configuración predeterminada de S3 Object Lambda.
2. Ejecute el siguiente comando `put-object` para cargar el paquete a un bucket de Amazon S3.

```
aws s3api put-object --bucket Amazon S3 bucket name --key
 s3objectlambda_deployment_package.zip --body release/
s3objectlambda_deployment_package.zip
```

3. Descargue la plantilla de AWS CloudFormation `s3objectlambda_defaultconfig.yaml` en la configuración predeterminada de S3 Object Lambda.
4. Ejecute el siguiente comando `deploy` para implementar la plantilla en su Cuenta de 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
```

Puede configurar esta plantilla AWS CloudFormation para que invoque a Lambda para operaciones `GET`, `HEAD` y `LIST` de la API. Para obtener más información sobre cómo modificar la configuración predeterminada de la plantilla, consulte the section called “Automatizar la configuración de S3 Object Lambda con AWS CloudFormation”.

Para crear un punto de acceso de Object Lambda mediante la AWS CLI

> **Note**
>
> Para utilizar los comandos siguientes, sustituya *`user input placeholders`* con su información.

En el siguiente ejemplo, se crea un punto de acceso de Object Lambda denominado *`my-object-lambda-ap`* para el bucket *`amzn-s3-demo-bucket1`* de la cuenta *`111122223333`*. En este ejemplo, se supone que ya se ha creado un punto de acceso estándar denominado *`example-ap`*. Para obtener información sobre la creación de un punto de acceso estándar, consulte the section called “Crear puntos de acceso”.

En este ejemplo, se utiliza la función compilada con anticipación `decompress` de AWS. Para obtener más información sobre funciones prediseñadas, consulte the section called “Uso de funciones creadas por AWS”.

1. Crear un bucket. En este ejemplo, usaremos *`amzn-s3-demo-bucket1`*. Para obtener más información acerca de cómo se crean los buckets, consulte the section called “Crear un bucket”.
2. Cree un punto de acceso estándar y adjúntelo a su bucket. En este ejemplo, usaremos *`example-ap`*. Para obtener información sobre la creación de puntos de acceso estándar, consulte the section called “Crear puntos de acceso”.
3. Realice una de las siguientes acciones siguientes:

   - Cree una función de Lambda en su cuenta que le gustaría utilizar para transformar su objeto de Amazon S). Para obtener más información sobre la creación de funciones de Lambda, consulte the section called “Escritura de funciones de Lambda”. Para usar la función

personalizada con AWS CLI, consulte Uso de Lambda con la AWS CLI en la Guía para desarrolladores de AWS Lambda.

- Utilice una función de Lambda de AWS creada con anticipación. Para obtener más información sobre funciones prediseñadas, consulte Uso de funciones de Lambda creadas por AWS.

4. Cree un archivo de configuración JSON denominado `my-olap-configuration.json`. En esta configuración, proporcione el punto de acceso de soporte y el nombre de recurso de Amazon (ARN) para la función de Lambda que creó en los pasos anteriores o el ARN de la función prediseñada que está utilizando.

   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. Ejecute el comando `create-access-point-for-object-lambda` para crear su punto de acceso de 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. (Opcional) Cree un archivo de política JSON denominado `my-olap-policy.json`.

   La adición de una política de recursos de punto de acceso Object Lambda puede controlar el uso del punto de acceso por recurso, usuario u otras condiciones. Esta política de recursos concede el permiso `GetObject` para la cuenta *444455556666* al punto de acceso de Object Lambda especificado.

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. (Opcional) Ejecute el comando `put-access-point-policy-for-object-lambda` para establecer la política de recursos.

```
aws s3control put-access-point-policy-for-object-lambda --account-id 111122223333
 --name my-object-lambda-ap --policy file://my-olap-policy.json
```

8. (Opcional) Especifique una carga.

   Una carga consiste en un JSON opcional que puede proporcionar a su función de AWS Lambda como entrada. Puede configurar cargas con diferentes parámetros para diferentes puntos de acceso Object Lambda que invoquen la misma función de Lambda, ampliando así la flexibilidad de la función de Lambda.

   La siguiente configuración del punto de acceso de Object Lambda muestra una carga con dos parámetros.

```
{
 "SupportingAccessPoint": "AccessPointArn",
 "CloudWatchMetricsEnabled": false,
 "TransformationConfigurations": [{
  "Actions": ["GetObject", "HeadObject", "ListObjects", "ListObjectsV2"],
  "ContentTransformation": {
   "AwsLambda": {
```

```
    "FunctionArn": "FunctionArn",
    "FunctionPayload": "{\"res-x\": \"100\",\"res-y\": \"100\"}"
   }
  }
 }]
}
```

La siguiente configuración del punto de acceso de Object Lambda muestra una carga con un parámetro y con `GetObject-Range`, `GetObject-PartNumber`, `HeadObject-Range` y `HeadObject-PartNumber` habilitados.

```
{
    "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

Cuando utilice los puntos de acceso Object Lambda, asegúrese de que la carga no contenga información confidencial.

Uso de la consola y la plantilla de AWS CloudFormation

Puede crear un punto de acceso de Object Lambda utilizando la configuración predeterminada proporcionada por Amazon S3. Puede descargar una plantilla de AWS CloudFormation y un código fuente de la función de Lambda del Repositorio GitHub e implementar estos recursos para configurar un punto de acceso de Object Lambda funcional.

Para obtener información sobre cómo modificar la configuración predeterminada de la plantilla de AWS CloudFormation, consulte the section called “Automatizar la configuración de S3 Object Lambda con AWS CloudFormation”.

Si quiere obtener información para configurar puntos de acceso Object Lambda mediante AWS CloudFormation sin la plantilla, consulte `AWS::S3ObjectLambda::AccessPoint` en la Guía del usuario de AWS CloudFormation.

Para cargar el paquete de implementación de la función de Lambda

1. Descargue el paquete de implementación de la función de AWS Lambda `s3objectlambda_deployment_package.zip` en la configuración predeterminada de S3 Object Lambda.
2. Cargue el paquete en un bucket de Amazon S3.

Para crear un punto de acceso de Object Lambda mediante la consola de AWS CloudFormation

1. Descargue la plantilla de AWS CloudFormation `s3objectlambda_defaultconfig.yaml` en la configuración predeterminada de S3 Object Lambda.
2. Inicie sesión en la consola de administración de AWS y abra la consola de AWS CloudFormation en https://console.aws.amazon.com/cloudformation.
3. Realice una de las siguientes acciones siguientes:
   - Si nunca ha usado AWS CloudFormation antes, en la página de inicio de AWS CloudFormation, seleccione Create stack (Crear pila).
   - Si ya ha utilizado AWS CloudFormation antes, en el panel de navegación izquierdo, elija Stacks (Pilas). Elija Create stack (Crear pila), y, a continuación, elija With new resources (standard) (Con nuevos recursos [estándar]).
4. En Prerequisite - Prepare template (Requisito previo - Preparar plantilla), elija Template is ready (La plantilla está lista).
5. En Specify template (Especificar plantilla), elija Upload a template file (Cargar un archivo de plantilla) y cargue `s3objectlambda_defaultconfig.yaml`.
6. Elija Siguiente.
7. En la página Specify stack details (Especificar detalles de la pila), ingrese un nombre para la pila.

8. En la sección Parameters (Parámetros), especifique los siguientes parámetros que se definen en la plantilla de la pila:

   a. Para CreateNewSupportingAccessPoint, realice una de las siguientes operaciones:

      - Si ya tiene un punto de acceso de soporte para el bucket de S3 en el que cargó la plantilla, elija false (falso).
      - Si desea crear un nuevo punto de acceso para este bucket, elija true (verdadero).

   b. Para EnableCloudWatchMonitoring, elija true o false, en función de si desea habilitar las alarmas y métricas de solicitud de Amazon CloudWatch.

   c. (Opcional) Para LambdaFunctionPayload, agregue el texto JSON que desea proporcionar a su función de Lambda como entrada. Puede configurar cargas con diferentes parámetros para diferentes puntos de acceso Object Lambda que invoquen la misma función de Lambda, ampliando así la flexibilidad de la función de Lambda.

      > **Important**
      >
      > Cuando utilice los puntos de acceso Object Lambda, asegúrese de que la carga no contenga información confidencial.

   d. Para LambdaFunctionRuntime, introduzca su tiempo de ejecución preferido para la función de Lambda. Las opciones disponibles son `nodejs14.x`, `python3.9` y `java11`.

   e. Para LambdaFunctionS3BucketName, introduzca el nombre del bucket de Amazon S3 donde cargó el paquete de implementación.

   f. Para LambdaFunctionS3Key, introduzca la clave de objeto de Amazon S3 en la que cargó el paquete de implementación

   g. Para LambdaFunctionS3ObjectVersion, introduzca la versión de objeto de Amazon S3 en la que cargó el paquete de implementación.

   h. Para ObjectLambdaAccessPointName, introduzca un nombre para el punto de acceso de Object Lambda.

   i. Para S3BucketName, introduzca el nombre del bucket de Amazon S3 que se asociará al punto de acceso de Object Lambda.

   j. Para SupportingAccessPointName, introduzca el nombre del punto de acceso de soporte.

> **Note**
>
> Este es un punto de acceso que está asociado al bucket de Amazon S3 que eligió en el paso anterior. Si no tiene ningún punto de acceso asociado al bucket de Amazon S3, puede configurar la plantilla para crear uno eligiendo true (verdadero) para CreateNewSupportingAccessPoint.

9. Elija Siguiente.
10. En la página Configurar opciones de pila, elija Siguiente.

    Para obtener más información sobre la configuración opcional en esta página, consulte Configuración de las opciones de pila de AWS CloudFormation en la Guía del usuario de AWS CloudFormation.
11. En la página Review (Revisar), elija Create stack (Crear pila).

Uso de la AWS Cloud Development Kit (AWS CDK)

Para obtener más información acerca de la configuración de puntos de acceso Object Lambda mediante el AWS CDK, consulte Biblioteca de constructos `AWS::S3ObjectLambda` en la Referencia de la API de AWS Cloud Development Kit (AWS CDK).

Uso de una plantilla de AWS CloudFormation para automatizar la configuración de S3 Object Lambda

Puede utilizar una plantilla de AWS CloudFormation para crear rápidamente un punto de acceso de Amazon S3 Object Lambda. La plantilla de CloudFormation crea automáticamente recursos relevantes, configura roles de AWS Identity and Access Management (IAM) y define una función de AWS Lambda que gestiona de manera automática las solicitudes a través del punto de acceso Object Lambda. Con la plantilla de CloudFormation, puede implementar prácticas recomendadas, mejorar su posición de seguridad y reducir los errores causados por los procesos manuales.

Este repositorio GitHub contiene la plantilla de CloudFormation y código fuente de la función de Lambda. Para obtener instrucciones sobre cómo utilizar la plantilla, consulte the section called “Creación de puntos de acceso Object Lambda”.

La función de Lambda proporcionada en la plantilla no ejecuta ninguna transformación. En su lugar, devuelve los objetos tal como están desde el bucket de S3. Puede clonar la función y agregar su

propio código de transformación para modificar y procesar los datos a medida que se devuelven a una aplicación. Para obtener más información sobre cómo modificar una función, consulte the section called “Modificación de la función de Lambda” y the section called “Escritura de funciones de Lambda”.

Modificación de la plantilla

Creación de un nuevo punto de acceso de soporte

S3 Object Lambda utiliza dos puntos de acceso: uno Object Lambda y otro S3 estándar, al que se denomina punto de acceso de apoyo. Cuando usted realiza una solicitud a un punto de acceso de Object Lambda, S3 invoca Lambda en su nombre o delega la solicitud al punto de acceso de soporte, según la configuración de S3 Object Lambda. Puede crear un nuevo punto de acceso de soporte pasando el siguiente parámetro como parte del comando `aws cloudformation deploy` al implementar la plantilla.

```
CreateNewSupportingAccessPoint=true
```

Configuración de una carga de funciones

Puede configurar una carga para proporcionar datos complementarios a la función de Lambda pasando el siguiente parámetro como parte del comando `aws cloudformation deploy` al implementar la plantilla.

```
LambdaFunctionPayload="format=json"
```

Habilitación del monitoreo de Amazon CloudWatch

Puede habilitar el monitoreo de CloudWatch pasando el siguiente parámetro como parte del comando `aws cloudformation deploy` al implementar la plantilla.

```
EnableCloudWatchMonitoring=true
```

Este parámetro habilita el punto de acceso de Object Lambda para las métricas de solicitudes de Amazon S3 y crea dos alarmas de CloudWatch para supervisar los errores del lado del cliente y del servidor.

> **Note**
>
> El uso de Amazon CloudWatch generará costos adicionales. Para obtener más información acerca de las métricas de CloudWatch para Amazon S3, consulte Monitorización y registro de puntos de acceso.
> Para obtener información sobre los precios, consulte Precios de CloudWatch.

Configuración de simultaneidad aprovisionada

Para reducir la latencia, puede configurar la simultaneidad aprovisionada para la función de Lambda que respalda el punto de acceso de Object Lambda editando la plantilla para incluir las siguientes líneas en `Resources`.

```
LambdaFunctionVersion:
      Type: AWS::Lambda::Version
      Properties:
        FunctionName: !Ref LambdaFunction
        ProvisionedConcurrencyConfig:
            ProvisionedConcurrentExecutions: Integer
```

> **Note**
>
> Se aplicarán cargos adicionales por la simultaneidad aprovisionada. Para obtener más información sobre la periodicidad aprovisionada, consulte Administración de la simultaneidad aprovisionada de Lambda en la Guía para desarrolladores de AWS Lambda.
> Para obtener más información sobre precios, consulte precios de AWS Lambda.

Modificación de la función de Lambda

Cambiar los valores del encabezado de una solicitud **`GetObject`**

De forma predeterminada, la función de Lambda reenvía todos los encabezados, excepto `Content-Length` y `ETag`, de la solicitud URL prefirmada al cliente `GetObject`. Según el código de transformación de la función de Lambda, puede elegir enviar nuevos valores de encabezado al cliente `GetObject`.

Puede actualizar la función de Lambda para enviar nuevos valores de encabezado pasándolos a la operación `WriteGetObjectResponse` de la API.

Por ejemplo, si la función de Lambda traduce el texto de los objetos de Amazon S3 a un idioma diferente, puede pasar un nuevo valor en el encabezado `Content-Language`. Para ello, modifique la función `writeResponse` como se indica a continuación.

```
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();
}
```

Para obtener una lista completa de los encabezados compatibles, consulte `WriteGetObjectResponse` en la Referencia de la API de Amazon Simple Storage Service.

Devolución de encabezados de metadatos

Puede actualizar la función de Lambda para enviar nuevos valores de encabezado pasándolos a la solicitud de la operación `WriteGetObjectResponse` de la API.

```
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();
}
```

Devolución de un nuevo código de estado

Puede devolver un código de estado personalizado al cliente `GetObject` pasándolo a la solicitud de la operación `WriteGetObjectResponse` de la API.

```
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();
}
```

Para obtener una lista completa de los códigos de estado compatibles, consulte `WriteGetObjectResponse` en la Referencia de la API de Amazon Simple Storage Service.

Aplicación de **`Range`** y **`partNumber`** al objeto de origen

De forma predeterminada, el punto de acceso de Object Lambda creado por la plantilla de CloudFormation puede gestionar parámetros `Range` y `partNumber`. La función de Lambda aplica el rango o el número de parte solicitado al objeto transformado. Para ello, la función debe descargar todo el objeto y ejecutar la transformación. En algunos casos, los rangos de objetos transformados pueden asignarse exactamente a los rangos de objetos de origen. Esto significa que solicitar un rango de bytes A-B en el objeto de origen y ejecutar la transformación puede producir el mismo resultado que solicitar todo el objeto, ejecutar la transformación y devolver el rango de bytes A-B en el objeto transformado.

En estos casos, puede cambiar la implementación de la función de Lambda para aplicar el rango o el número de parte directamente al objeto de origen. Este método reduce la latencia general de la función y la memoria requerida. Para obtener más información, consulte the section called “Trabajar con encabezados Range y partNumber”.

Deshabilitación de **`Range`** y gestión de **`partNumber`**

De forma predeterminada, el punto de acceso de Object Lambda creado por la plantilla de CloudFormation puede gestionar parámetros `Range` y `partNumber`. Si no necesita este comportamiento, puede desactivarlo eliminando las siguientes líneas de la plantilla:

```
AllowedFeatures:
  - GetObject-Range
  - GetObject-PartNumber
  - HeadObject-Range
  - HeadObject-PartNumber
```

Transformación de objetos grandes

De forma predeterminada, la función de Lambda procesa todo el objeto en la memoria antes de que pueda comenzar a transmitir la respuesta a S3 Object Lambda. Puede modificar la función para transmitir la respuesta a medida que realiza la transformación. Hacer esto ayuda a reducir la latencia de transformación y el tamaño de la memoria de la función de Lambda. Para ver un ejemplo de implementación, consulte el Ejemplo de contenido comprimido de streaming.

Uso de puntos de acceso de Amazon S3 Object Lambda

La realización de solicitudes a través de puntos de acceso de Amazon S3 Object Lambda funciona de igual modo que las solicitudes a través de otros puntos de acceso. Para obtener más información acerca de cómo realizar solicitudes a través de un punto de acceso, consulte Usar puntos de acceso. Puede realizar solicitudes a través de los puntos de acceso de Object Lambda mediante la consola, la AWS Command Line Interface (AWS CLI), los SDK de AWS o la API de REST de Amazon S3.

> **Important**
>
> Los nombres de recurso de Amazon (ARN) para los puntos de acceso de Object Lambda utilizan un nombre de servicio `s3-object-lambda`. Por lo tanto, los ARN del punto de acceso de Object Lambda comienzan con `arn:aws:s3-object-lambda` en lugar de con `arn:aws:s3`, que se utiliza con otros puntos de acceso.

Cómo encontrar el ARN para su punto de acceso de Object Lambda

Para utilizar un punto de acceso de Object Lambda con la AWS CLI o los SDK de AWS debe conocer el nombre de recurso de Amazon (ARN) del punto de acceso de Object Lambda. En los siguientes ejemplos, se muestra cómo encontrar el ARN para un punto de acceso de Object Lambda mediante la consola de Amazon S3 o la AWS CLI.

Uso de la consola de S3

Para encontrar el ARN para su punto de acceso de Object Lambda mediante la consola

1. Inicie sesión AWS Management Console Management Console y abra la consola de Amazon S3 en https://console.aws.amazon.com/s3/.
2. En el panel de navegación del lado izquierdo de la consola, elija Object Lambda Access Points (Puntos de acceso de Object Lambda).
3. Seleccione el botón de opción situado junto al punto de acceso de Object Lambda cuyo ARN desea copiar.
4. Seleccionar Copy ARN (Copiar ARN).

Uso de la AWS CLI

Para encontrar el ARN para su punto de acceso de Object Lambda mediante la AWS CLI

1. Para recuperar una lista de los puntos de acceso de Object Lambda asociados a su Cuenta de AWS, ejecute el siguiente comando. Antes de ejecutar el comando, reemplace el ID de cuenta *111122223333* con el ID de su Cuenta de AWS.

```
aws s3control list-access-points-for-object-lambda --account-id 111122223333
```

2. Revise el resultado del comando para encontrar el ARN del punto de acceso de Object Lambda que desea utilizar. El resultado del comando anterior tendrá un aspecto semejante al del siguiente ejemplo.

```
{
    "ObjectLambdaAccessPointList": [
        {
            "Name": "my-object-lambda-ap",
            "ObjectLambdaAccessPointArn": "arn:aws:s3-object-lambda:us-
east-1:111122223333:accesspoint/my-object-lambda-ap"
```

```
        },
        ...
    ]
}
```

Cómo usar un alias de estilo de bucket para su punto de acceso de Object Lambda de bucket de S3

Al crear un punto de acceso de Object Lambda, Amazon S3 genera automáticamente un alias único para el punto de acceso de Object Lambda. Puede utilizar este alias en lugar de un nombre de bucket de Amazon S3 o el nombre de recurso de Amazon (ARN) del punto de acceso de Object Lambda en las operaciones de plano de datos de punto de acceso. Para obtener una lista de las operaciones, consulte Compatibilidad del punto de acceso con servicios de AWS.

Se crea un nombre de alias de punto de acceso de Object Lambda en el mismo espacio de nombres que un bucket de Amazon S3. Este nombre de alias se genera de forma automática y no se puede cambiar. Para un punto de acceso de Object Lambda existente, se asigna le automáticamente un alias. Un nombre de alias de punto de acceso de Object Lambda cumple con todos los requisitos de un nombre de bucket válido de Amazon S3 y consta de las siguientes partes:

Object Lambda Access Point name prefix-*metadata*--ol-s3

> **Note**
>
> El sufijo `--ol-s3` está reservado para los nombres de alias de punto de acceso de Object Lambda y no se puede utilizar para los nombres de punto de acceso de Object Lambda o de bucket. Para obtener más información acerca de las reglas de nomenclatura del bucket de Amazon S3, consulte Reglas de nomenclatura de buckets.

Los siguientes ejemplos muestran un ARN y un alias de punto de acceso de Object Lambda para un punto de acceso de Object Lambda llamado *my-object-lambda-access-point*:

- ARN: `arn:aws:s3-object-lambda:`*region*`:`*account-id*`:accesspoint/`*my-object-lambda-access-point*
- Alias de punto de acceso de Object Lambda: *my-object-lambda-acc*-*1a4n8yjrb3kda96f67zwrwiiuse1a*--ol-s3

Al utilizar un punto de acceso de Object Lambda, puede utilizar el nombre del alias del punto de acceso de Object Lambda sin que tenga que realizar cambios importantes en el código.

Al eliminar un punto de acceso de Object Lambda, el nombre del alias del punto de acceso de Object Lambda queda inactivo y desaprovisionado.

Cómo encontrar el alias para su punto de acceso de Object Lambda

Uso de la consola de S3

Para encontrar el alias para su punto de acceso de Object Lambda mediante la consola

1. Inicie sesión AWS Management Console Management Console y abra la consola de Amazon S3 en https://console.aws.amazon.com/s3/.
2. En el panel de navegación del lado izquierdo de la consola, elija Object Lambda Access Points (Puntos de acceso de Object Lambda).
3. Para el punto de acceso de Object Lambda que desea utilizar, copie el valor de Alias del punto de acceso del objeto Lambda.

Uso de la AWS CLI

Al crear un punto de acceso de Object Lambda, Amazon S3 genera de forma automática un nombre de alias de punto de acceso de Object Lambda, tal como se muestra en el siguiente ejemplo. Para ejecutar este comando, sustituya los *user input placeholders* con su propia información. Para obtener información acerca de cómo crear un punto de acceso de Object Lambda mediante la AWS CLI, consulte Para crear un punto de acceso de Object Lambda mediante la 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"
    }
}
```

El nombre del alias del punto de acceso de Object Lambda generado tiene dos campos:

- El campo `Value` es el valor del alias del punto de acceso de Object Lambda.
- El campo `Status` es el estado del alias del punto de acceso de Object Lambda. Si el estado es `PROVISIONING`, Amazon S3 está aprovisionando el alias del punto de acceso de Object Lambda y el alias aún no se puede usar. Si el estado es `READY`, el alias del punto de acceso de Object Lambda se habrá aprovisionado correctamente y ya se podrá usar.

Para obtener más información sobre el tipo de dato `ObjectLambdaAccessPointAlias` en la API de REST, consulte CreateAccessPointForObjectLambda y ObjectLambdaAccessPointAlias en la Referencia de la API de Amazon Simple Storage Service.

Cómo usar el alias del punto de acceso de Object Lambda

Puede utilizar un alias de punto de acceso de Object Lambda en lugar de un nombre de bucket de Amazon S3 para las operaciones que se enumeran en Compatibilidad del punto de acceso con servicios de AWS.

En el siguiente ejemplo de la AWS CLI para el comando `get-bucket-location` se usa el alias del punto de acceso del bucket para devolver la Región de AWS en la que está el bucket. Para ejecutar este comando, sustituya los *`user input placeholders`* con su propia información.

```
aws s3api get-bucket-location --bucket my-object-lambda-
acc-w7i37nq6xuzgax3jw3oqtifiusw2a--ol-s3

{
    "LocationConstraint": "us-west-2"
}
```

Si el alias del punto de acceso de Object Lambda de una solicitud no es válido, se devuelve el código de error `InvalidAccessPointAliasError`. Para obtener más información sobre `InvalidAccessPointAliasError`, consulte Lista de códigos de error en la Referencia de la API de Amazon Simple Storage Service.

Las limitaciones de un alias de punto de acceso de Object Lambda son las mismas que las de un alias de punto de acceso. Para obtener más información acerca de las limitaciones de un alias de punto de acceso, consulte Limitaciones.

Consideraciones de seguridad para los puntos de acceso de S3 Object Lambda

Con Amazon S3 Object Lambda S3 puede realizar transformaciones personalizadas en los datos a medida que salen de Amazon S3 utilizando la escala y la flexibilidad de AWS Lambda como plataforma de computación. S3 y Lambda permanecen seguros por defecto, pero para mantener esta seguridad, se requiere una consideración especial por parte del autor de la función de Lambda. S3 Object Lambda requiere que todo el acceso se realice a través de entidades autenticadas (sin acceso anónimo) y a través de HTTPS.

Para mitigar los riesgos de seguridad, recomendamos lo siguiente:

- Amplíe el rol de ejecución de Lambda al conjunto de permisos más pequeño posible.
- Siempre que sea posible, asegúrese de que su función de Lambda acceda a Amazon S3 a través de la URL prefirmada proporcionada.

Configuración de políticas de IAM

Los puntos de acceso de S3 admiten políticas de recursos de AWS Identity and Access Management (IAM) que permiten controlar el uso del punto de acceso en función del recurso, del usuario o de otras condiciones. Para obtener más información, consulte Configuración de las políticas de IAM para puntos de acceso de Object Lambda.

Comportamiento cifrado

Dado que el punto de acceso de Object Lambda utiliza Amazon S3 y AWS Lambda, existen diferencias en cuanto al comportamiento de cifrado. Para obtener más información sobre el comportamiento cifrado predeterminado de S3, consulte Establecer el comportamiento del cifrado predeterminado del lado del servidor para los buckets de Amazon S3.

- Cuando se utiliza el cifrado del lado del servidor de S3 con puntos de acceso de Object Lambda, el objeto se descifra antes de enviarlo a Lambda. Una vez que el objeto se envía a Lambda, se procesa sin cifrar (en el caso de una solicitud `GET` o `HEAD`).
- Para evitar que se registre la clave de cifrado, S3 rechaza solicitudes `GET` y `HEAD` de objetos que se cifran utilizando el cifrado del lado del servidor con claves proporcionadas por el cliente (SSE-C). Sin embargo, la función de Lambda podría recuperar estos objetos si tiene acceso a la clave proporcionada por el cliente.

- Al utilizar el cifrado del cliente de S3 con puntos de acceso de Object Lambda, asegúrese de que Lambda tenga acceso a la clave de cifrado para poder descifrar y volver a cifrar el objeto.

Seguridad en puntos de acceso

S3 Object Lambda utiliza dos puntos de acceso: uno Object Lambda y otro S3 estándar, al que se denomina punto de acceso de apoyo. Cuando usted realiza una solicitud a un punto de acceso de Object Lambda, S3 invoca Lambda en su nombre o delega la solicitud al punto de acceso de soporte, según la configuración de S3 Object Lambda. Cuando se invoca Lambda para una solicitud, S3 genera una URL prefirmada para su objeto en su nombre a través del punto de acceso de soporte. Su función de Lambda recibe la URL como entrada cuando se invoca.

Puede configurar su función de Lambda para que use la URL prefirmada para recuperar el objeto original en lugar de invocar S3 directamente. Al utilizar este modelo, puede aplicar mejores límites de seguridad a los objetos. Puede limitar el acceso directo a objetos a través de buckets de S3 o puntos de acceso de S3 a un conjunto limitado de usuarios o roles de IAM. Este método también protege sus funciones de Lambda de estar sujetas al problema del suplente confuso, donde una función mal configurada con permisos diferentes a su invocador podría permitir o denegar el acceso a objetos cuando no debería.

Acceso público al punto de acceso de Object Lambda

S3 Object Lambda no permite el acceso anónimo ni público porque Amazon S3 debe autorizar su identidad para completar cualquier solicitud de S3 Object Lambda. Al invocar solicitudes a través de un punto de acceso de Object Lambda, necesita el permiso `lambda:InvokeFunction` para la función de Lambda configurada. Del mismo modo, al invocar otras operaciones de la API a través de un punto de acceso de Object Lambda, debe tener los permisos `s3:*` necesarios.

Sin estos permisos, las solicitudes para invocar Lambda o delegar a S3 fallarán con errores HTTP 403 (Prohibido). Todos los accesos deben ser realizados por entidades autenticadas. Si necesita un acceso público, puede utilizar Lambda@Edge como posible alternativa. Para obtener más información, consulte Personalización en el borde con Lambda@Edge en la guía para desarrolladores de Amazon CloudFront.

Direcciones IP del punto de acceso de Object Lambda

Las subredes `describe-managed-prefix-lists` admiten puntos de conexión de nube privada virtual (VPC) de puerta de enlace y están relacionadas con la tabla de enrutamiento de los puntos de

conexión de VPC. Dado que el punto de acceso de Object Lambda no admite la VPC de puerta de enlace, faltan sus rangos de IP. Los rangos que faltan pertenecen a Amazon S3, pero no se admiten en los puntos de conexión de VPC de la puerta de enlace. Para obtener más información acerca de `describe-managed-prefix-lists`, consulte DescribeManagedPrefixLists en la Referencia de la API de Amazon EC2 y Rangos de direcciones IP de AWS en la Referencia general de AWS.

Configuración de las políticas de IAM para puntos de acceso de Object Lambda

Los puntos de acceso de Amazon S3 admiten políticas de recursos de AWS Identity and Access Management que le permiten controlar el uso del punto de acceso en función del recurso, del usuario o de otras condiciones. Puede controlar el acceso mediante una política de recursos opcional en su punto de acceso de Object Lambda o una política de recursos en el punto de acceso de apoyo. Para obtener un ejemplo paso a paso, consulte Tutorial: transformación de datos para su aplicación con S3 Object Lambda y Tutorial: detección y redacción de datos de PII con S3 Object Lambda y Amazon Comprehend.

Los siguientes cuatro recursos deben contar con los permisos correspondientes para trabajar con puntos de acceso de Object Lambda:

- La identidad de IAM, como el usuario o el rol. Para obtener más información acerca de las identidades y prácticas recomendadas de IAM, consulte Identidades (usuarios, grupos de usuarios y roles) de IAM en la Guía del usuario de IAM.
- El bucket y el punto de acceso estándar asociado Cuando se trabaja con puntos de acceso Object Lambda, este punto de acceso estándar se conoce como punto de acceso de apoyo.
- El punto de acceso de Object Lambda.
- La función de AWS Lambda

> **Important**
>
> Antes de guardar la política, asegúrese de resolver advertencias de seguridad, errores, advertencias generales y sugerencias de AWS Identity and Access Management Access Analyzer. IAM Access Analyzer ejecuta verificaciones de política para validarla contra la Gramática de la política de IAM y las prácticas recomendadas. Estas verificaciones generan hallazgos y proporcionan recomendaciones procesables para ayudarlo a crear políticas funcionales y que se ajustan a las prácticas recomendadas de seguridad.
> Para obtener más información sobre la validación de políticas utilizando IAM Access Analyzer, consulte Validación de políticas de IAM Access Analyzer en la Guía del usuario

de IAM. Para ver una lista de advertencias, errores y sugerencias que devuelve IAM Access Analyzer, consulte Referencia de verificación de políticas de IAM Access Analyzer.

En los siguientes ejemplos de política se supone que dispone de los siguientes recursos:

- Un bucket de Amazon S3 con el siguiente Nombre de recurso de Amazon (ARN):

  `arn:aws:s3:::amzn-s3-demo-bucket1`
- Un punto de acceso estándar de Amazon S3 en este bucket con el siguiente ARN:

  `arn:aws:s3:us-east-1:111122223333:accesspoint/my-access-point`
- Un punto de acceso de Object Lambda con el siguiente ARN:

  `arn:aws:s3-object-lambda:us-east-1:111122223333:accesspoint/my-object-lambda-ap`
- Una función de AWS Lambda con el siguiente ARN:

  `arn:aws:lambda:us-east-1:111122223333:function:MyObjectLambdaFunction`

> **Note**
>
> Si utiliza una función de Lambda desde la cuenta, debe incluir la versión de la función específica en la declaración de política. En el siguiente ARN de ejemplo, la versión se indica mediante *1*:
> `arn:aws:lambda:us-east-1:111122223333:function:MyObjectLambdaFunction:1`
> Lambda no admite la agregación de políticas de IAM a la versión $LATEST. Para obtener más información acerca de las funciones de Lambda, consulte Versiones de la función de Lambda en la Guía para desarrolladores de AWS Lambda.

Example - Política de bucket que delega el control de acceso a los puntos de acceso estándar

El siguiente ejemplo de política de bucket de S3 delega el control de acceso de un bucket a los puntos de acceso estándar del bucket. Esta política permite el acceso completo a todos los puntos de acceso de la cuenta del propietario del bucket. Por lo tanto, todo el acceso a este bucket está controlado por las políticas asociadas a sus puntos de acceso. Los usuarios pueden leer desde el

bucket solo a través de un punto de acceso de S3, lo que significa que las operaciones se pueden invocar solo a través de puntos de acceso. Para obtener más información, consulte Delegar el control de acceso a los puntos de acceso.

```
{
    "Version": "2012-10-17",
    "Statement" : [
    {
        "Effect": "Allow",
        "Principal" : { "AWS":"account-ARN"},
        "Action" : "*",
        "Resource" : [
            "arn:aws:s3:::amzn-s3-demo-bucket1",
            "arn:aws:s3:::amzn-s3-demo-bucket1/*"
        ],
        "Condition": {
            "StringEquals" : { "s3:DataAccessPointAccount" : "Bucket owner's account
 ID" }
        }
    }]
}
```

Example – Política de IAM que otorga al usuario los permisos necesarios para usar un punto de acceso de Object Lambda

La siguiente política de IAM otorga permisos de usuario a la función de Lambda, al punto de acceso estándar y al punto de acceso de 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"
    }
  ]
}
```

Habilitar permisos para roles de ejecución de Lambda

Cuando se realizan solicitudes `GET` a un punto de acceso de Object Lambda, su función de Lambda necesita permiso para enviar datos al punto de acceso de S3 Object Lambda. Este permiso se proporciona activando el permiso `s3-object-lambda:WriteGetObjectResponse` en el rol de ejecución de la función de Lambda. Puede crear un nuevo rol de ejecución o actualizar un rol existente.

> **Note**
>
> Su función solo necesita el permiso `s3-object-lambda:WriteGetObjectResponse` si realiza una solicitud `GET`.

Para crear un rol de ejecución en la consola de IAM

1. Abra la consola de IAM en https://console.aws.amazon.com/iam/.
2. En el panel de navegación izquierdo, seleccione Roles.
3. Elija Crear rol.
4. En Common use cases (Casos de uso comunes), elija Lambda.
5. Elija Siguiente.
6. En la página Add permissions (Añadir permisos), busque la política administrada por AWS `AmazonS3ObjectLambdaExecutionRolePolicy` y, a continuación, seleccione la casilla de verificación situada junto al nombre de la política.

   Esta política debe contener la acción `s3-object-lambda:WriteGetObjectResponse`.
7. Elija Siguiente.
8. En la página Name, review, and create (Nombre, revisar, creer) de Role name (Nombre de rol), introduzca **`s3-object-lambda-role`**.
9. (Opcional) Añada una descripción y etiquetas para este rol.
10. Elija Create role (Crear rol).
11. Aplique el recurso recién creado **`s3-object-lambda-role`** como rol de ejecución de la función de Lambda. Esto se puede hacer durante o después de la creación de la función de Lambda en la consola de Lambda.

Para obtener más información sobre los roles de ejecución, consulte Rol de ejecución de Lambda en la Guía para desarrolladores de AWS Lambda.

Uso de claves de contexto con puntos de acceso de Object Lambda

S3 Object Lambda evaluará claves de contexto tales como `s3-object-lambda:TlsVersion` o `s3-object-lambda:AuthType` relacionadas con la conexión o la firma de la solicitud. Amazon S3 evalúa todas las demás claves de contexto, como `s3:prefix`.

Soporte para puntos de acceso de Object Lambda

Cuando S3 Object Lambda recibe una solicitud de un navegador o la solicitud incluye un encabezado `Origin`, S3 Object Lambda siempre añade un campo de encabezado `“AllowedOrigins":"*"`.

Para obtener más información, consulte Uso compartido de recursos entre orígenes (CORS).

Escritura de funciones de Lambda para puntos de acceso de S3 Object Lambda

En esta sección se detalla cómo escribir las funciones de AWS Lambda para usarlas con puntos de acceso de Amazon S3 Object Lambda.

Para obtener información sobre procedimientos integrales completos para algunas tareas de S3 Object Lambda, consulte:

- Tutorial: transformación de datos para su aplicación con S3 Object Lambda
- Tutorial: detección y redacción de datos de PII con S3 Object Lambda y Amazon Comprehend
- Tutorial: Uso de S3 Object Lambda para agregar marcas de agua dinámicas a las imágenes a medida que se recuperan

Temas

- Trabajar con solicitudes GetObject en Lambda
- Trabajar con solicitudes HeadObject en Lambda
- Trabajar con solicitudes ListObjects en Lambda
- Trabajar con solicitudes ListObjectsV2 en Lambda
- Formato y uso del contexto del evento
- Trabajar con encabezados Range y partNumber

Trabajar con solicitudes **`GetObject`** en Lambda

En esta sección se asume que el punto de acceso de Object Lambda está configurado para llamar a la función de Lambda para `GetObject`. S3 Object Lambda incluye la operación de la API de Amazon S3, `WriteGetObjectResponse`, que le permite a la función de Lambda proporcionar datos personalizados y encabezados de respuesta al intermediario de `GetObject`.

`WriteGetObjectResponse` proporciona un amplio control sobre el código de estado, los encabezados de respuesta y el cuerpo de respuesta, en función de sus necesidades de procesamiento. Puede utilizar `WriteGetObjectResponse` para responder con todo el objeto transformado, partes del objeto transformado u otras respuestas en función del contexto de la aplicación. En la siguiente sección se muestran ejemplos únicos del uso de la operación de la API `WriteGetObjectResponse`.

- Ejemplo 1: responder con un código de estado HTTP 403 (Prohibido)
- Ejemplo 2: responder con una imagen transformada
- Ejemplo 3: transmitir contenido comprimido

Ejemplo 1: responder con un código de estado HTTP 403 (Prohibido)

Puede usar `WriteGetObjectResponse` para responder con el código de estado HTTP 403 (Prohibido) en función del contenido del objeto.

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 };
}
```

Ejemplo 2: responder con una imagen transformada

Cuando realice una transformación de imagen, es posible que necesite todos los bytes del objeto de origen antes de comenzar a procesarlos. En este caso, la solicitud `WriteGetObjectResponse` devolverá el objeto completo a la aplicación solicitante en una llamada.

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 };
}
```

Ejemplo 3: transmitir contenido comprimido

Cuando se comprimen objetos, los datos comprimidos se producen de forma progresiva. En consecuencia, la solicitud `WriteGetObjectResponse` se puede utilizar para devolver los datos comprimidos tan pronto como estén listos. Como se muestra en el siguiente ejemplo, no es necesario conocer la duración de la transformación completada.

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**
>
> Aunque S3 Object Lambda permite hasta 60 segundos para enviar una respuesta completa al intermediario a través de la solicitud `WriteGetObjectResponse`, el tiempo disponible podría ser menor. Por ejemplo, el tiempo de espera de la función de Lambda puede ser inferior a 60 segundos. En otros casos, el intermediario puede tener tiempos de espera más estrictos.

Para que el intermediario original reciba una respuesta que no sea el código de estado HTTP 500 (Error interno del servidor), la llamada de `WriteGetObjectResponse` debe completarse. Si la función de Lambda realiza la devolución, mediante con una excepción o de otro modo, antes de que se llame a la API `WriteGetObjectResponse`, el intermediario original recibirá una respuesta 500 (Error interno del servidor). Las excepciones lanzadas durante el tiempo que se tarda en completar la respuesta dan como resultado respuestas truncadas al intermediario. Si la función de Lambda recibe un código de estado de HTTP 200 (OK) de la llamada a la API `WriteGetObjectResponse`, entonces el intermediario original ha enviado la solicitud completa. La respuesta de la función de Lambda, independientemente de si se produce una excepción o no, es ignorada por S3 Object Lambda.

Cuando se llama a la operación de API `WriteGetObjectResponse`, Amazon S3 requiere la ruta y el token de solicitud del contexto del evento. Para obtener más información, consulte Formato y uso del contexto del evento.

Los parámetros de la ruta y del token de solicitud son necesarios para conectar la respuesta de `WriteGetObjectResult` con el intermediario original. Aunque siempre es conveniente reintentar las respuestas 500 (Error interno del servidor), debido a que el token de solicitud es un token de un solo uso, los intentos posteriores de utilizarlo podrían dar lugar a respuestas de código de estado

HTTP 400 (Solicitud incorrecta). Aunque la llamada a `WriteGetObjectResponse` con la ruta los tokens de solicitud no necesariamente debe realizarse desde la función de Lambda invocada, sí debe realizarla una identidad de la misma cuenta. La llamada también debe completarse antes de que la función de Lambda finalice la ejecución.

Trabajar con solicitudes **`HeadObject`** en Lambda

En esta sección se asume que el punto de acceso de Object Lambda está configurado para llamar a la función de Lambda para `HeadObject`. Lambda recibirá una carga JSON que contiene una clave llamada `headObjectContext`. Dentro del contexto, hay una sola propiedad llamada `inputS3Url`, que es una URL prefirmada para el punto de acceso de soporte para `HeadObject`.

La URL prefirmada incluirá las siguientes propiedades si se especifican:

- `versionId` (en los parámetros de la consulta)
- `requestPayer` (en el encabezado `x-amz-request-payer`)
- `expectedBucketOwner` (en el encabezado `x-amz-expected-bucket-owner`)

Otras propiedades no estarán prefirmadas y, por lo tanto, no se incluirán. Las opciones no firmadas que se envían como encabezados se pueden añadir manualmente a la solicitud al llamar a la URL prefirmada que se encuentra en los encabezados `userRequest`. No se admiten las opciones de cifrado del lado del servidor para `HeadObject`.

Para conocer los parámetros URI de la sintaxis de la solicitud, consulte `HeadObject` en la Referencia de la API de Amazon Simple Storage Service

En el siguiente ejemplo, se muestra una carga de entrada JSON de Lambda para `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"
}
```

La función de Lambda debe devolver un objeto JSON que contenga los encabezados y los valores que se devolverán para la llamada `HeadObject`.

En el siguiente ejemplo se muestra la estructura del JSON de la respuesta de Lambda para `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>
        ...
    };
}
```

El siguiente ejemplo muestra cómo utilizar la URL prefirmada para rellenar la respuesta modificando los valores del encabezado según sea necesario antes de devolver el 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
        }
```

Trabajar con solicitudes **`ListObjects`** en Lambda

En esta sección se asume que el punto de acceso de Object Lambda está configurado para llamar a la función de Lambda para `ListObjects`. Lambda recibirá la carga JSON con un nuevo objeto llamado `listObjectsContext`. `listObjectsContext` contiene una sola propiedad, `inputS3Url`, que es una URL prefirmada para el punto de acceso de soporte para `ListObjects`.

A diferencia de `GetObject` y `HeadObject`, la URL prefirmada incluirá las siguientes propiedades si se especifican:

- Todos los parámetros de la consulta
- `requestPayer` (en el encabezado `x-amz-request-payer`)
- `expectedBucketOwner` (en el encabezado `x-amz-expected-bucket-owner`)

Para conocer los parámetros URI de la sintaxis de la solicitud, consulte `ListObjects` en la Referencia de la API de Amazon Simple Storage Service

> **Important**
>
> Recomendamos que utilice la versión más reciente, ListObjectSv2, cuando desarrolle aplicaciones. Para la compatibilidad con versiones anteriores, Amazon S3 sigue siendo compatible con `ListObjects`.

En el siguiente ejemplo, se muestra la carga de entrada JSON de Lambda para `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"
}
```

La función de Lambda debe devolver un objeto JSON que contenga el código de estado, el resultado XML de la lista o la información de error que devolverá el objeto S3 Lambda.

S3 Object Lambda no procesa ni valida `listResultXml`, pero en su lugar lo reenvía al intermediario de `ListObjects`. Para `listBucketResult`, S3 Object Lambda espera que determinadas propiedades sean de un tipo específico y generará excepciones si no puede analizarlas. No se pueden proporcionar `listResultXml` y `listBucketResult` al mismo tiempo.

En el siguiente ejemplo se muestra cómo usar la URL prefirmada para llamar a Amazon S3 y usar el resultado para rellenar una respuesta, incluida la comprobación de errores.

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
```

En el siguiente ejemplo se muestra la estructura del JSON de la respuesta de Lambda para `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'
        },
        ...
        ],
    }
}
```

Trabajar con solicitudes **`ListObjectsV2`** en Lambda

En esta sección se asume que el punto de acceso de Object Lambda está configurado para llamar a la función de Lambda para `ListObjectsV2`. Lambda recibirá la carga JSON con un nuevo objeto llamado `listObjectsV2Context`. `listObjectsV2Context` contiene una sola propiedad, `inputS3Url`, que es una URL prefirmada para el punto de acceso de soporte para `ListObjectsV2`.

A diferencia de `GetObject` y `HeadObject`, la URL prefirmada incluirá las siguientes propiedades si se especifican:

- Todos los parámetros de la consulta
- `requestPayer` (en el encabezado `x-amz-request-payer`)
- `expectedBucketOwner` (en el encabezado `x-amz-expected-bucket-owner`)

Para conocer los parámetros URI de la sintaxis de la solicitud, consulte `ListObjectsV2` en la Referencia de la API de Amazon Simple Storage Service

En el siguiente ejemplo, se muestra la carga de entrada JSON de Lambda para `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"
}
```

La función de Lambda debe devolver un objeto JSON que contenga el código de estado, el resultado XML de la lista o la información de error que devolverá el objeto S3 Lambda.

S3 Object Lambda no procesa ni valida `listResultXml`, pero en su lugar lo reenvía al intermediario de `ListObjectsV2`. Para `listBucketResult`, S3 Object Lambda espera que

determinadas propiedades sean de un tipo específico y generará excepciones si no puede analizarlas. No se pueden proporcionar `listResultXml` y `listBucketResult` al mismo tiempo.

En el siguiente ejemplo se muestra cómo usar la URL prefirmada para llamar a Amazon S3 y usar el resultado para rellenar una respuesta, incluida la comprobación de errores.

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
```

En el siguiente ejemplo se muestra la estructura del JSON de la respuesta de Lambda para `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'
            },
        ...
        ],
    }
}
```

Formato y uso del contexto del evento

Amazon S3 Object Lambda proporciona contexto sobre la solicitud que se realiza en el evento transferido a la función de AWS Lambda. A continuación se muestra un ejemplo de solicitud: Las descripciones de los campos se incluyen después del ejemplo.

```
{
    "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"
}
```

Los siguientes campos están incluidos en la solicitud:

- `xAmzRequestId`: ID de solicitud de Amazon S3 para esta solicitud. Le recomendamos que registre este valor para ayudar con la depuración.
- `getObjectContext`: detalles de entrada y salida de las conexiones a Amazon S3 y S3 Object Lambda.
  - `inputS3Url`: URL prefirmada que se puede utilizar para obtener el objeto original de Amazon S3. La URL está firmada utilizando la identidad del intermediario que la llamó originalmente, y los permisos de ese usuario se aplicarán cuando se utilice la URL. Si hay encabezados firmados en la URL, la función de Lambda debe incluir estos encabezados los en la llamada a Amazon S3, excepto en el caso del encabezado `Host`.
  - `outputRoute`: token de enrutamiento que se agrega a la URL de S3 Object Lambda cuando la función de Lambda llama a `WriteGetObjectResponse`.
  - `outputToken`: token opaco utilizado por S3 Object Lambda para hacer coincidir la llamada `WriteGetObjectResponse` con el intermediario original.
- `configuration`: información de configuración sobre el punto de acceso de Object Lambda.
  - `accessPointArn`: nombre de recurso de Amazon (ARN) del punto de acceso de Object Lambda que ha recibido esta solicitud.
  - `supportingAccessPointArn`: ARN del punto de acceso de apoyo especificado en la configuración del punto de acceso de Object Lambda.
  - `payload`: datos personalizados que se aplican a la configuración del punto de acceso de Object Lambda. S3 Object Lambda trata estos datos como una cadena opaca, por lo que es posible que deba decodificarse antes de su uso.
- `userRequest`: información sobre la llamada original a S3 Object Lambda.
  - `url`: URL decodificada de la solicitud recibida por S3 Object Lambda, excluyendo cualquier parámetro de consulta relacionado con la autorización.
  - `headers`: un mapa de cadena a cadenas que contienen los encabezados HTTP y los valores de la llamada original, excluyendo cualquier encabezado relacionado con la autorización. Si el mismo encabezado aparece varias veces, los valores de cada instancia del mismo encabezado se combinan en una lista delimitada por comas. Las mayúsculas y minúsculas de los encabezados originales se conservan en este mapa.
- `userIdentity`: detalles sobre la identidad que hizo la llamada a S3 Object Lambda. Para obtener más información, consulte Registro de eventos de datos para registros de seguimiento en la Guía del usuario de AWS CloudTrail.

  - `type`: tipo de identidad.
  - `accountId`: Cuenta de AWS a la que pertenece la identidad.
  - `userName`: nombre descriptivo de la identidad que realizó la llamada.
  - `principalId`: identificador único para la identidad que ha efectuado la llamada.
  - `arn`: ARN de la entidad principal que hizo la llamada. La última sección del ARN contiene el usuario o el rol que realizó la llamada.
  - `sessionContext`: si la solicitud se realizó con credenciales de seguridad temporales, este elemento proporciona información sobre la sesión que se creó para esas credenciales.
  - `invokedBy`: nombre del Servicio de AWS que realizó la solicitud, como Amazon EC2 Auto Scaling o AWS Elastic Beanstalk.
  - `sessionIssuer`: si la solicitud se realizó con credenciales de seguridad temporales, este elemento proporciona información acerca de cómo se obtuvieron las credenciales.
- `protocolVersion`: ID de versión del contexto proporcionado. El formato de este campo es `{Major Version}.{Minor Version}`. Los números de versión secundaria son siempre números de dos dígitos. Cualquier eliminación o cambio en la semántica de un campo precisa un bache de versión principal y requiere la opción opcional activa. Amazon S3 puede agregar nuevos campos en cualquier momento, momento en el que podría experimentar un problema de versión secundaria. Por la naturaleza de las implementaciones de software, puede que vea varias versiones secundarias en uso a la vez.

Trabajar con encabezados Range y partNumber

Cuando se trabaja con objetos grandes en Amazon S3 Object Lambda, puede utilizar el encabezado HTTP `Range` para descargar un rango de bytes especificado de un objeto. Para obtener diferentes rangos de bytes dentro del mismo objeto, puede utilizar conexiones concurrentes a Amazon S3. También puede usar el parámetro `partNumber` (número entero entre 1 y 10 000) que realiza una solicitud de rango para la parte especificada del objeto.

Puesto que hay varias formas de gestionar una solicitud que incluya los parámetros `Range` o `partNumber`, S3 Object Lambda no aplica estos parámetros al objeto transformado. En su lugar, la función de AWS Lambda debe implementar esta funcionalidad según sea necesario para la aplicación.

Para usar los parámetros `Range` y `partNumber` con S3 Object Lambda, haga lo siguiente:

- Active estos parámetros en la configuración del punto de acceso de Object Lambda.

- Escriba una función de Lambda que pueda gestionar las solicitudes que incluyan estos parámetros.

Esta operación se describe en los siguientes pasos.

Paso 1: Configurar el punto de acceso de Object Lambda

De forma predeterminada, los puntos de acceso de Object Lambda responden con un código de estado HTTP 501 (No implementado) a cualquier solicitud `GetObject` o `HeadObject` que contenga un parámetro `Range` o `partNumber`, ya sea en los encabezados o en los parámetros de consulta.

Para permitir que un punto de acceso de Object Lambda acepte este tipo de solicitudes, debe incluir `GetObject-Range`, `GetObject-PartNumber`, `HeadObject-Range` o `HeadObject-PartNumber` en la sección `AllowedFeatures` de la configuración de su punto de acceso de Object Lambda. Para obtener más información acerca de la actualización de la configuración del punto de acceso de Object Lambda, consulte Creación de puntos de acceso Object Lambda.

Paso 2: Implementar el tratamiento de **`Range`** o **`partNumber`** en la función de Lambda

Cuando el punto de acceso de Object Lambda invoque la función de Lambda con una solicitud `GetObject` o `HeadObject` de rango, los parámetros `Range` o `partNumber` se incluyen en el contexto del evento. La ubicación del parámetro en el contexto del evento depende del parámetro que se haya utilizado y de cómo se incluyó en la solicitud original al punto de acceso de Object Lambda, tal como se explica en la tabla siguiente.

Parámetro	Ubicación en el contexto del evento
`Range` (encabezado)	`userRequest.headers.Range`
`Range` (parámetro de consulta)	`userRequest.url` (parámetro de consulta `Range`)
`partNumber`	`userRequest.url` (parámetro de consulta `partNumber`)

> **Important**
>
> La URL prefirmada proporcionada para su punto de acceso de Object Lambda no contiene el parámetro `Range` o `partNumber` de la solicitud original. Consulte las siguientes opciones sobre cómo manejar estos parámetros en la función de AWS Lambda.

Después de extraer el valor de `Range` o `partNumber`, puede adoptar uno de los siguientes enfoques en función de las necesidades de su aplicación:

A. Asigne el **`Range`** o **`partNumber`** solicitados al objeto transformado (recomendado).

La forma más fiable de gestionar solicitudes `Range` o `partNumber` es hacer lo siguiente:

- Recupere el objeto completo de Amazon S3.
- Transforme el objeto.
- Aplique los parámetros `Range` o `partNumber` solicitados al objeto transformado.

Para ello, utilice la URL prefirmada proporcionada para obtener el objeto completo de Amazon S3 y, a continuación, procesarlo según sea necesario. Para ver un ejemplo de función de Lambda que procesa así un parámetro `Range`, consulte este ejemplo en el repositorio AWS de GitHub.

B. Asigne el **`Range`** solicitado a la URL prefirmada.

En algunos casos la función de Lambda puede asignar el `Range` solicitado directamente a la URL prefirmada para recuperar solo parte del objeto de Amazon S3. Este enfoque es adecuado solo si la transformación cumple los dos criterios siguientes:

1. La función de transformación se puede aplicar a rangos de objetos parciales.
2. La aplicación del parámetro `Range` antes o después de la función de transformación da como resultado el mismo objeto transformado.

Por ejemplo, una función de transformación que convierte en mayúsculas todos los caracteres de un objeto codificado en ASCII cumple los dos criterios anteriores. La transformación se puede aplicar a parte de un objeto y aplicar el parámetro `Range` antes de la transformación da el mismo resultado que aplicar el parámetro después de la transformación.

Por el contrario, una función que invierte los caracteres de un objeto codificado en ASCII no cumple estos criterios. Dicha función cumple el criterio 1, ya que se puede aplicar a rangos de objetos parciales. No obstante, no cumple el criterio 2, porque cuando se aplica el parámetro

Range de la transformación se obtienen resultados distintos que si se aplica el parámetro después de la transformación.

Considere una solicitud para aplicar la función a los tres primeros caracteres de un objeto con el contenido abcdefg. Aplicar el parámetro Range antes de la transformación solo recupera abc y luego invierte los datos, devolviendo cba. Pero si el parámetro se aplica después de la transformación, la función recupera todo el objeto, lo invierte y, a continuación, aplica el parámetro Range, devolviendo gfe. Puesto que los resultados son diferentes, esta función no debe aplicar el parámetro Range cuando se recupera el objeto de Amazon S3. En su lugar, debe recuperar el objeto completo, realizar la transformación y solo después aplicar el parámetro Range.

> **Warning**
>
> En muchos casos si se aplica el parámetro Range a la URL prefirmada, el resultado será un comportamiento inesperado por parte de la función de Lambda o el cliente solicitante. A menos que tenga la seguridad de que su aplicación funcionará correctamente cuando se recupere solo un objeto parcial de Amazon S3, recomendamos que recupere y transforme objetos completos como se describió anteriormente en el enfoque A.

Si la solicitud cumple los criterios descritos anteriormente en este método B, puede simplificar la función AWS Lambda recuperando solo el intervalo de objetos solicitado y, a continuación, ejecutando la transformación en ese intervalo.

En el siguiente ejemplo de código Java, se muestra cómo realizar lo siguiente:

- Recupere el encabezado Range de la solicitud GetObject.
- Añada el encabezado Range a la URL prefirmada que Lambda puede utilizar para recuperar el rango solicitado desde 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;
```

```
}
```

Uso de funciones de Lambda creadas por AWS

AWS proporciona algunas funciones de AWS Lambda predefinidas que puede utilizar con Amazon S3 Object Lambda para detectar y redactar información de identificación personal (PII) y descomprimir objetos de S3. Estas funciones de Lambda están disponibles en AWS Serverless Application Repository. Puede seleccionar estas funciones a través de la AWS Management Console cuando se crea el punto de acceso de Object Lambda.

Para obtener más información sobre cómo implementar aplicaciones sin servidor desde AWS Serverless Application Repository, consulte Implementación de aplicaciones en la Guía para desarrolladores de AWS Serverless Application Repository.

> **Note**
>
> Los siguientes ejemplos solo se pueden usar con solicitudes `GetObject`.

Ejemplo 1: control de acceso PII

Esta función de Lambda utiliza Amazon Comprehend, un servicio de procesamiento de lenguaje natural (NLP) que usa machine learning para encontrar información y relaciones en textos. Esta función detecta automáticamente información de identificación personal (PII), como nombres, direcciones, fechas, números de tarjeta de crédito y números de seguridad social en los documentos de su bucket de Amazon S3. Si tiene documentos en el bucket que incluyen PII, puede configurar la función de control de acceso PII para detectar estos tipos de entidades PII y restringir el acceso a usuarios no autorizados.

Para empezar, implemente la siguiente función de Lambda en su cuenta y agregue el nombre de recurso de Amazon (ARN) para la función a la configuración de su punto de acceso de Object Lambda.

A continuación se muestra un ejemplo de ARN para un :

```
arn:aws:serverlessrepo:us-east-1:111122223333:applications/
ComprehendPiiAccessControlS3ObjectLambda
```

Puede agregar la vista de esta función en la AWS Management Console mediante el siguiente enlace de AWS Serverless Application Repository: ComprehendPiiAccessControlS3ObjectLambda.

Para ver esta función en GitHub, consulte Amazon Comprehend S3 Object Lambda.

Ejemplo 2: redacción de PII

Esta función de Lambda utiliza Amazon Comprehend, un servicio de procesamiento de lenguaje natural (NLP) que usa machine learning para encontrar información y relaciones en textos. Esta función redacta automáticamente información de identificación personal (PII), como nombres, direcciones, fechas, números de tarjeta de crédito y números de seguridad social de los documentos de su bucket de Amazon S3.

Si tiene documentos en su bucket que incluyen información como números de tarjeta de crédito o información de cuenta bancaria, puede configurar la función PII Redaction S3 Object Lambda para detectar PII y luego devolver una copia de estos documentos en los que se redactan los tipos de entidad PII.

Para empezar, implemente la siguiente función de Lambda en su cuenta y agregue el ARN para la función a la configuración de su punto de acceso de Object Lambda.

A continuación se muestra un ejemplo de ARN para un :

```
arn:aws:serverlessrepo:us-east-1:111122223333::applications/
ComprehendPiiRedactionS3ObjectLambda
```

Puede agregar la vista de esta función en la AWS Management Console mediante el siguiente enlace de AWS Serverless Application Repository: ComprehendPiiRedactionS3ObjectLambda.

Para ver esta función en GitHub, consulte Amazon Comprehend S3 Object Lambda.

Para obtener información sobre procedimientos integrales completos para algunas tareas de S3 Object Lambda en redacción de PII, consulte Tutorial: detección y redacción de datos de PII con S3 Object Lambda y Amazon Comprehend.

Ejemplo 3: descompresión

La función de Lambda `S3ObjectLambdaDecompression` puede descomprimir objetos almacenados en Amazon S3 en uno de los seis formatos de archivo comprimidos: `bzip2`, `gzip`, `snappy`, `zlib`, `zstandard` y `ZIP`.

Para empezar, implemente la siguiente función de Lambda en su cuenta y agregue el ARN para la función a la configuración de su punto de acceso de Object Lambda.

A continuación se muestra un ejemplo de ARN para esta función:

```
arn:aws:serverlessrepo:us-east-1:111122223333::applications/S3ObjectLambdaDecompression
```

Puede agregar la vista de esta función en la AWS Management Console mediante el siguiente enlace de AWS Serverless Application Repository: S3ObjectLambdaDecompression.

Para ver esta función en GitHub vea Descompresión de S3 Object Lambda.

Prácticas recomendadas y directrices para S3 Object Lambda

Cuando utilice S3 Object Lambda, siga estas prácticas recomendadas y directrices para optimizar las operaciones y el rendimiento.

Temas

- Trabajo con S3 Object Lambda
- Servicios de AWS utilizados en relación con S3 Object Lambda
- Encabezados Range y partNumber
- Transformar la expiry-date
- Trabajar con la AWS CLI y los SDK de AWS

Trabajo con S3 Object Lambda

S3 Object Lambda solo admite el procesamiento de solicitudes `GET`, `LIST` y `HEAD`. Cualquier otra solicitud no invoca AWS Lambda y, en su lugar, devuelve respuestas de API estándar y no transformadas. Puede crear un máximo de 1000 puntos de acceso de Object Lambda por Cuenta de AWS por cada región. La función de AWS Lambda que utilice debe estar en la misma Cuenta de AWS y región que el punto de acceso de Object Lambda.

S3 Object Lambda permite hasta 60 segundos para transmitir una respuesta completa al intermediario. Su función también está sujeta a cuotas predeterminadas de AWS Lambda. Para obtener más información, consulte Cuotas de Lambda en la Guía para desarrolladores de AWS Lambda.

Cuando S3 Object Lambda invoca la función de Lambda especificada, y usted es responsable de asegurarse de que cualquier dato sobrescrito o eliminado de Amazon S3 por la función o aplicación de Lambda especificada sea intencional y correcto.

Solo se puede utilizar S3 Object Lambda para realizar operaciones en objetos. No se pueden utilizar S3 Object Lambda para realizar otras operaciones de Amazon S3, como modificar o eliminar buckets. Para obtener una lista completa de las operaciones de S3 que admiten puntos de acceso, consulte Compatibilidad de los puntos de acceso con las operaciones de S3.

Además de esta lista, los puntos de acceso de Object Lambda no admiten las operaciones de la API `POST Object`, `CopyObject` (como origen) y `SelectObjectContent`.

Servicios de AWS utilizados en relación con S3 Object Lambda

S3 Object Lambda conecta Amazon S3, AWS Lambda y, opcionalmente, otros Servicios de AWS que usted elija a fin de entregar objetos relevantes para las aplicaciones que hagan las solicitudes. Todos los Servicios de AWS utilizados con S3 Object Lambda se rigen por sus respectivos acuerdos de nivel de servicio (SLA). Por ejemplo, si algún Servicio de AWS no cumple con su compromiso de servicio, usted puede recibir un crédito de servicio conforme con lo estipulado en el acuerdo de nivel de servicio correspondiente.

Encabezados **`Range`** y **`partNumber`**

Cuando se trabaja con objetos grandes, puede utilizar el encabezado HTTP `Range` para descargar un rango de bytes especificado de un objeto. Cuando usa el encabezado `Range`, su solicitud solo obtiene la parte especificada del objeto. También puede usar el encabezado `partNumber` para realizar una solicitud de rango para la parte especificada desde el objeto.

Para obtener más información, consulte, Trabajar con encabezados Range y partNumber.

Transformar la **`expiry-date`**

Puede abrir o descargar objetos transformados desde su punto de acceso de Object Lambda en la AWS Management Console. Estos objetos no deben estar vencidos. Si la función de Lambda transforma la `expiry-date` de los objetos, es posible que vea objetos caducados que no se pueden abrir ni descargar. Este comportamiento se aplica únicamente a los objetos restaurados de S3 Glacier Flexible Retrieval y S3 Glacier Deep Archive.

Trabajar con la AWS CLI y los SDK de AWS

Los subcomandos de S3 de la AWS Command Line Interface (AWS CLI) (`cp`,`mv` y `sync`) y el uso de la clase AWS SDK for Java de `TransferManager` no están permitidos para su uso con S3 Object Lambda.

Tutoriales de S3 Object Lambda

Los siguientes tutoriales presentan procedimientos integrales completos para algunas tareas comunes de S3 Object Lambda.

- Tutorial: transformación de datos para su aplicación con S3 Object Lambda
- Tutorial: detección y redacción de datos de PII con S3 Object Lambda y Amazon Comprehend
- Tutorial: Uso de S3 Object Lambda para agregar marcas de agua dinámicas a las imágenes a medida que se recuperan

Depuración de S3 Object Lambda

Las solicitudes de puntos de acceso Amazon S3 Object Lambda pueden dar como resultado una nueva respuesta de error cuando algo sale mal con la invocación o ejecución de la función de Lambda. Estos errores siguen el mismo formato que los errores estándar de Amazon S3. Para obtener información acerca de los errores de S3 Object Lambda, consulte la Lista de códigos de error de S3 Object Lambda en la Referencia de API de Amazon Simple Storage Service.

Para obtener más información sobre la depuración general de funciones de Lambda, consulte Monitoreo y solución de problemas de aplicaciones de Lambda en la Guía para desarrolladores de AWS Lambda.

Para obtener información sobre los errores estándar de Amazon S3, consulte Respuestas de errores en la Referencia de API de Amazon Simple Storage Service.

Puede activar métricas de solicitud en Amazon CloudWatch para los puntos de acceso de Object Lambda. Estas métricas ayudan a supervisar el rendimiento operativo de su punto de acceso. Puede activar métricas de solicitud durante o después de la creación del punto de acceso de Object Lambda. Para obtener más información, consulte Métricas de solicitud de S3 Object Lambda en CloudWatch.

Para obtener un registro más detallado de las solicitudes realizadas a los puntos de acceso de Object Lambda, puede activar los eventos de datos de AWS CloudTrail. Para obtener más

información, consulte Registro de eventos de datos para registros de seguimiento en la Guía del usuario de AWS CloudTrail.

Para tutoriales de S3 Object Lambda, consulte lo siguiente:

- Tutorial: transformación de datos para su aplicación con S3 Object Lambda
- Tutorial: detección y redacción de datos de PII con S3 Object Lambda y Amazon Comprehend
- Tutorial: Uso de S3 Object Lambda para agregar marcas de agua dinámicas a las imágenes a medida que se recuperan

Para obtener más información acerca de los puntos de acceso estándar, consulte Administración del acceso a datos con puntos de acceso de Amazon S3.

Para obtener información sobre cómo trabajar con buckets, consulte Descripción general de los buckets. Para obtener información acerca del uso de objetos, consulte Información general de los objetos de Amazon S3.

¿Qué es S3 Express One Zone?

Amazon S3 Express One Zone es una clase de almacenamiento de Amazon S3 en zona única de alto rendimiento que está diseñada específicamente para ofrecer acceso constante a los datos en milisegundos de un solo dígito para los datos a los que accede para las aplicaciones sensibles a la latencia. S3 Express One Zone es la clase de almacenamiento de objetos en la nube con la latencia más baja disponible en la actualidad, con una velocidad de acceso a los datos hasta 10 veces más rápida y unos costos de solicitud un 50 % más bajos que los de S3 Standard. Las aplicaciones pueden beneficiarse inmediatamente de que las solicitudes se completen en un orden de magnitud más rápido. S3 Express One Zone ofrece una elasticidad de rendimiento similar a la de otras clases de almacenamiento de S3.

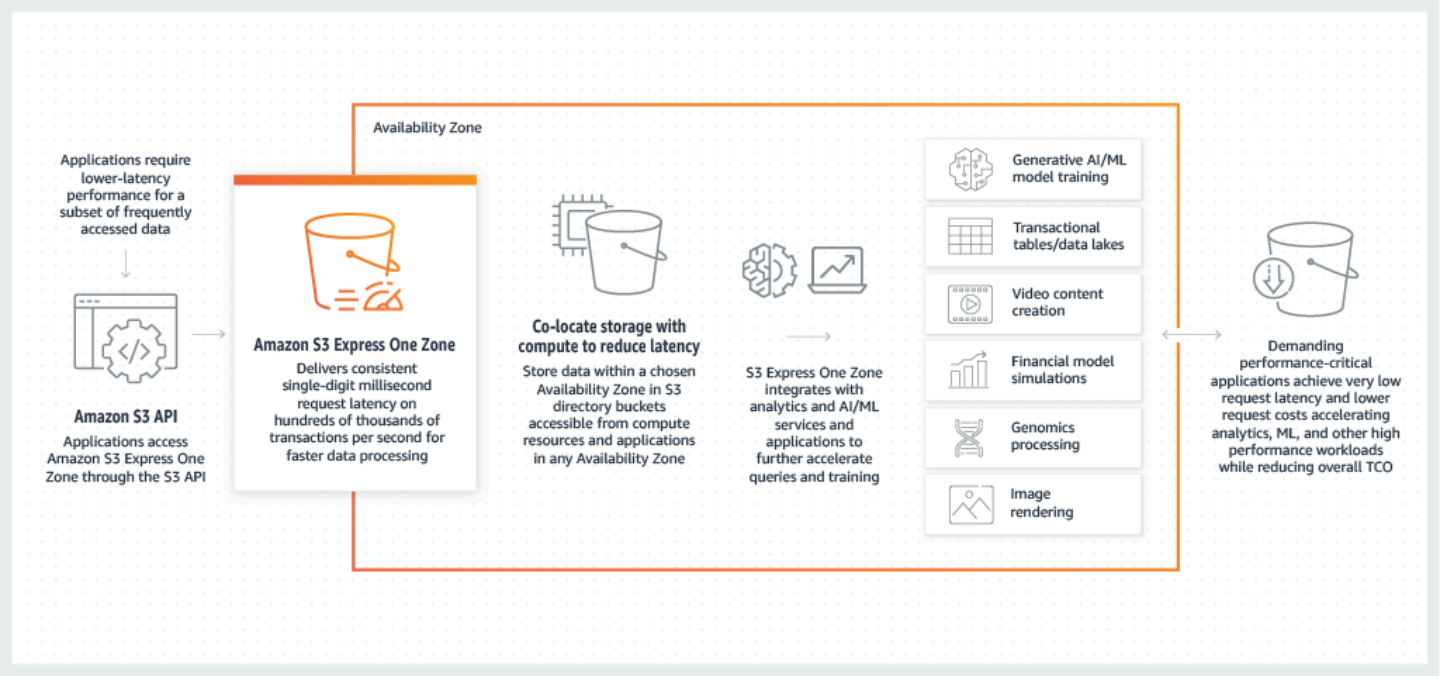
De igual modo que con otras clases de almacenamiento de Amazon S3, no es necesario planificar ni aprovisionar los requisitos de capacidad o rendimiento por adelantado. Puede ampliar o reducir su almacenamiento, según sus necesidades, y acceder a sus datos a través de la API de Amazon S3.

S3 Express One Zone es la primera clase de almacenamiento de S3 en la que se puede seleccionar una única zona de disponibilidad con la opción de coubicar su almacenamiento de objetos junto con sus recursos informáticos, lo que brinda la mayor velocidad de acceso posible. Además, para aumentar aún más la velocidad de acceso y admitir cientos de miles de solicitudes por segundo, los datos de la clase de almacenamiento S3 Express One Zone se almacenan en un nuevo tipo de bucket: un bucket de directorio de Amazon S3. Cada bucket de directorio puede admitir cientos de miles de transacciones por segundo (TPS), independientemente de los nombres de las claves o del patrón de acceso.

La clase de almacenamiento Amazon S3 Express One Zone está diseñada para ofrecer una disponibilidad del 99,95 % dentro de una única zona de disponibilidad y está respaldada por el contrato de nivel de servicio de Amazon S3. Con S3 Express One Zone, sus datos se almacenan de forma redundante en varios dispositivos dentro de una única zona de disponibilidad. S3 Express One Zone está diseñado para controlar los fallos simultáneos de dispositivos mediante la detección y la reparación rápidas de cualquier redundancia perdida. Si el dispositivo existente detecta un fallo, S3 Express One Zone transfiere automáticamente las solicitudes a los nuevos dispositivos dentro de una zona de disponibilidad. Esta redundancia ayuda a garantizar el acceso ininterrumpido a los datos dentro de una zona de disponibilidad.

S3 Express One Zone es ideal para cualquier aplicación en la que sea importante minimizar la latencia necesaria para acceder a un objeto. Estas aplicaciones pueden ser flujos de trabajo

interactivos entre humanos, como la edición de vídeo, en los que los profesionales creativos necesitan un acceso fluido al contenido desde sus interfaces de usuario. S3 Express One Zone también beneficia a las cargas de trabajo de análisis y machine learning que tienen requisitos de capacidad de respuesta similares a los de sus datos, especialmente las cargas de trabajo con muchos accesos más pequeños o un gran número de accesos aleatorios. S3 Express One Zone se puede usar con otros Servicios de AWS para admitir cargas de trabajo de análisis, inteligencia artificial y machine learning (IA y ML), como Amazon EMR, Amazon SageMaker y Amazon Athena.



Al usar S3 Express One Zone, puede interactuar con su bucket de directorio en una nube privada virtual (VPC) mediante un punto de conexión de VPC de puerta de enlace. Los puntos de conexión de puerta de enlace permiten acceder a buckets de directorio de S3 Express One Zone desde su VPC sin necesidad de una puerta de enlace de Internet ni de un dispositivo NAT para la VPC, y sin costo adicional.

Con los buckets de directorio puede usar muchas de las operaciones y características de la API de Amazon S3 que usa con los buckets de uso general y otras clases de almacenamiento. Estos incluyen Mountpoint para Amazon S3, cifrado del servidor con claves administradas por Amazon S3 (SSE-S3), Operaciones por lotes de S3 y S3 Block Public Access. También puede acceder a S3 Express One Zone desde la consola de Amazon S3, la AWS Command Line Interface (AWS CLI), los SDK de AWS y la API de REST de Amazon S3.

Para obtener más información acerca de S3 Express One Zone, consulte los siguientes temas.

- Información general

Información general

Para optimizar el rendimiento y reducir la latencia, S3 Express One Zone presenta los siguientes conceptos nuevos.

Una sola zona de disponibilidad

La clase de almacenamiento Amazon S3 Express One Zone está diseñada para ofrecer una disponibilidad del 99,95 % dentro de una única zona de disponibilidad y está respaldada por el contrato de nivel de servicio de Amazon S3. Con S3 Express One Zone, sus datos se almacenan de forma redundante en varios dispositivos dentro de una única zona de disponibilidad. S3 Express One Zone está diseñado para controlar los fallos simultáneos de dispositivos mediante la detección y la reparación rápidas de cualquier redundancia perdida. Si el dispositivo existente detecta un fallo, S3 Express One Zone transfiere automáticamente las solicitudes a los nuevos dispositivos dentro de una zona de disponibilidad. Esta redundancia ayuda a garantizar el acceso ininterrumpido a los datos dentro de una zona de disponibilidad.

Una zona de disponibilidad consiste en uno o varios centros de datos discretos con alimentación, redes y conectividad redundantes en una Región de AWS. Al crear un bucket de directorio, debe elegir la zona de disponibilidad y la Región de AWS donde se colocará el bucket.

Buckets de directorio

Existen dos tipos de buckets de Amazon S3: buckets de uso general de S3 y buckets de directorio de S3. Los buckets de uso general son el tipo de bucket predeterminado de Amazon S3 que se utiliza en la gran mayoría de los casos de uso de S3. Los buckets de directorio utilizan únicamente la clase de almacenamiento S3 Express One Zone, que está diseñada para cargas de trabajo o aplicaciones fundamentales para el rendimiento que requieren una latencia uniforme en milisegundos de un solo dígito. Elija el tipo de bucket que mejor se adapte a sus requisitos de rendimiento y aplicación.

Los buckets de directorio organizan los datos jerárquicamente en directorios, a diferencia de la estructura de almacenamiento plana de los buckets de uso general. No hay límites de prefijos para los buckets de directorio y los directorios individuales pueden realizar un escalado horizontal.

Los buckets de directorio utilizan la clase de almacenamiento S3 Express One Zone, que está diseñada para que la utilicen aplicaciones sensibles al rendimiento. Con S3 Express One Zone puede seleccionar una única zona de disponibilidad con la opción de coubicar el almacenamiento de objetos junto con los recursos informáticos, lo que brinda la mayor velocidad de acceso posible. A diferencia de los buckets de uso general, que almacenan objetos de forma redundante en varias zonas de disponibilidad en Regiones de AWS.

Para obtener más información acerca de estos buckets de directorio, consulte Buckets de directorio. Para obtener más información acerca de los buckets de uso general, consulte Descripción general de los buckets.

Puntos de conexión y puntos de conexión de VPC de puerta de enlace

Las operaciones de la API de administración para buckets de directorio están disponibles a través de un punto de conexión regional y se denominan operaciones de la API de puntos de conexión regionales. Algunos ejemplos de operaciones de la API de puntos de conexión regionales son `CreateBucket` y `DeleteBucket`. Tras crear un bucket de directorio, puede utilizar las operaciones de la API de puntos de conexión zonales para cargar y administrar los objetos de su bucket de directorio. Las operaciones de la API de puntos de conexión zonales están disponibles a través de un punto de conexión zonal. Algunos ejemplos de operaciones de la API de puntos de conexión zonales son `PutObject` y `CopyObject`.

Puede acceder a S3 Express One Zone desde la VPC mediante los puntos de conexión de VPC de puerta de enlace. Después de crear un punto de conexión de la puerta de enlace, puede agregarlo como destino en la tabla de enrutamiento para el tráfico destinado desde la VPC a S3 Express One Zone. Al igual que con Amazon S3, el uso de puntos de conexión de puertas de enlace no conlleva ningún cargo adicional. Para obtener más información acerca de cómo configurar los puntos de conexión de VPC de puerta de enlace, consulte Redes para S3 Express One Zone.

Autorización basada en sesiones

Con S3 Express One Zone, puede autenticar y autorizar las solicitudes mediante un nuevo mecanismo basado en sesiones, que está optimizado para ofrecer la latencia más baja. Puede utilizar `CreateSession` para solicitar credenciales temporales que otorgan acceso de baja latencia a su bucket. Estas credenciales temporales tienen el alcance de un bucket de directorio de S3 específico. Los tokens de sesión se utilizan solo con operaciones (de nivel de objeto) zonales (con la excepción de CopyObject). Para obtener más información, consulte Autorización de `CreateSession`.

Los SDK de AWS compatibles con S3 Express One Zone controlan el establecimiento y la actualización de la sesión en su nombre. Para proteger sus sesiones, las credenciales de seguridad temporales caducan a los 5 minutos. Esto significa que puede empezar a usar las operaciones de la API inmediatamente después de descargar e instalar los SDK de AWS y configurar los permisos de AWS Identity and Access Management (IAM) necesarios.

Características de S3 Express One Zone

Las siguientes características de S3 están disponibles para S3 Express One Zone. Para obtener una lista completa de las operaciones de la API compatibles y las características no compatibles, consulte ¿En qué se diferencia S3 Express One Zone?.

Gestión de acceso y seguridad

Con los buckets de directorio, puede utilizar las siguientes características para auditar y administrar el acceso. De forma predeterminada, los buckets de directorio son privados y solo pueden acceder a ellos los usuarios a los que se concede explícitamente el acceso. A diferencia de los buckets de uso general, que pueden establecer el límite de control de acceso en el nivel de bucket, prefijo o etiqueta de objeto, el límite de control de acceso de los buckets de directorio se establece únicamente en el nivel de bucket. Para obtener más información, consulte AWS Identity and Access Management (IAM) para S3 Express One Zone.

- S3 Block Public Access: todas las configuraciones de S3 Block Public Access están habilitadas de forma predeterminada en el nivel de bucket. Esta configuración predeterminada no se puede modificar.
- S3 Object Ownership (Aplicada al propietario del bucket es la opción predeterminada): las listas de control de acceso (ACL) no son compatibles con los buckets de directorio. Los buckets de directorio utilizan automáticamente la configuración Aplicada al propietario del bucket en la S3 Object Ownership. Aplicada al propietario del bucket significa que las ACL están desactivadas y el propietario del bucket tiene automáticamente la propiedad y el control total sobre cada objeto del bucket. Esta configuración predeterminada no se puede modificar.
- AWS Identity and Access Management (IAM): IAM ayuda a controlar de forma segura el acceso a los buckets de directorio. Puede utilizar IAM para conceder acceso a las operaciones de la API de administración de buckets (regionales) y a las operaciones de la API de administración de objetos (zonales) mediante la acción `s3express:CreateSession`. Para obtener más información, consulte AWS Identity and Access Management (IAM) para S3 Express One Zone. A diferencia de

las acciones de administración de objetos, las acciones de administración de buckets no pueden ser entre cuentas. Solo el propietario del bucket puede realizar esas acciones.

- Políticas de buckets: utilice el lenguaje de políticas basado en IAM para configurar permisos basados en recursos para sus buckets de directorio. También puede usar IAM para controlar el acceso a la operación de la API `CreateSession`, que le permite usar las operaciones de la API de administración de objetos o zonales. Puede conceder acceso a la misma cuenta o entre cuentas a las operaciones de la API zonales. Para obtener más información acerca de las políticas y los permisos de S3 Express One Zone, consulte AWS Identity and Access Management (IAM) para S3 Express One Zone.
- Analizador de acceso de IAM para S3: para evaluar y monitorear sus políticas de acceso, asegúrese de que las políticas solo proporcionen el acceso previsto a sus recursos de S3.

Registro y monitorización

S3 Express One Zone usa las siguientes herramientas de registro y monitoreo de S3 que puede utilizar para monitorear y controlar cómo se utilizan sus recursos:

- Métricas de Amazon CloudWatch: para monitorear sus recursos y aplicaciones de AWS, utilice CloudWatch para recopilar y realizar el seguimiento de métricas. S3 Express One Zone utiliza el mismo espacio de nombres de CloudWatch que otras clases de almacenamiento de Amazon S3 (`AWS/S3`) y admite métricas de almacenamiento diarias para los buckets de directorio: `BucketSizeBytes` y `NumberOfObjects`. Para obtener más información, consulte Monitorización de métricas con Amazon CloudWatch.
- Registros de AWS CloudTrail: AWS CloudTrail es un Servicio de AWS que lo ayuda a implementar la auditoría de riesgos y operaciones, la gobernanza y el cumplimiento de su Cuenta de AWS mediante el registro de las acciones realizadas por un usuario, un rol o un Servicio de AWS. Para S3 Express One Zone, CloudTrail captura las operaciones de la API de puntos de conexión regionales (por ejemplo, `CreateBucket` y `PutBucketPolicy`) como eventos de administración y operaciones de API zonales (por ejemplo, `GetObject` y `PutObject`) como eventos de datos. Estos eventos incluyen las acciones realizadas en la AWS Management Console, la AWS Command Line Interface (AWS CLI), los SDK de AWS y las operaciones de la API de AWS. Para obtener más información, consulte Registro con AWS CloudTrail para S3 Express One Zone.

> **Note**
>
> S3 Express One Zone no admite registros de acceso al servidor de Amazon S3.

Administración de objetos

Tras crear un bucket de directorio, puede administrar el almacenamiento de objetos mediante la consola de Amazon S3, los SDK de AWS y AWS CLI. Las siguientes características están disponibles para la administración de objetos con S3 Express One Zone:

- Operaciones por lotes de S3: utilice las operaciones por lotes para realizar operaciones masivas en objetos de buckets de directorio, por ejemplo, Copiar y la función Invoke AWS Lambda. Por ejemplo, puede usar las operaciones por lotes para copiar objetos entre buckets de directorio y buckets de uso general. Con Operaciones por lotes, puede administrar miles de millones de objetos a escala con una sola solicitud de S3 mediante los SDK de AWS o la AWS CLI, o con unos pocos clics en la consola de Amazon S3.
- Importación: después de crear un bucket de directorio, puede rellenar el bucket con objetos mediante la característica de importación de la consola de Amazon S3. Importar es un método simplificado para crear trabajos de Batch Operations a fin de copiar objetos de buckets de uso general a buckets de directorio.

SDK de AWS y bibliotecas de clientes

Después de crear un bucket de directorio y cargar un objeto en el bucket, puede administrar el almacenamiento de objetos de la siguiente manera.

- Mountpoint para Amazon S3: Mountpoint para Amazon S3 es un cliente de archivos de código abierto que ofrece acceso de alto rendimiento, lo que reduce los costos de computación de los lagos de datos en Amazon S3. Mountpoint para Amazon S3 traduce las llamadas a la API del sistema de archivos local en llamadas a la API de objetos de S3, como `GET` y `LIST`. Es ideal para cargas de trabajo de lagos de datos de lectura intensiva que procesan petabytes de datos y necesitan el alto rendimiento elástico proporcionado por Amazon S3 para escalar vertical y horizontalmente en miles de instancias.
- S3A: S3A es una interfaz compatible con Hadoop recomendada para acceder a los almacenes de datos en Amazon S3. S3A reemplaza al cliente de sistema de archivos S3N Hadoop.

- PyTorch en AWS: PyTorch en AWS es un marco de aprendizaje profundo de código abierto que facilita el desarrollo de modelos de machine learning y su implementación en producción.
- SDK de AWS: puede utilizar los SDK de AWS para desarrollar aplicaciones con Amazon S3. Los SDK de AWS simplifican las tareas de programación dado que incluyen la API de REST de Amazon S3 subyacente. Para obtener más información sobre el uso de los SDK de AWS con S3 Express One Zone, consulte the section called “SDK de AWS”.

Cifrado y protección de datos

Los objetos almacenados en buckets de directorio se almacenan mediante cifrado del servidor con claves administradas por Amazon S3 (SSE-S3). Los buckets de directorio no admiten el cifrado del servidor con claves de AWS Key Management Service (AWS KMS) (SSE-KMS), el cifrado del servidor con claves de cifrado proporcionadas por el cliente (SSE-C) ni el cifrado del servidor de doble capa con AWS KMS keys (DSSE-KMS). Para obtener más información, consulte Protección y cifrado de datos y Uso del cifrado del servidor con claves administradas por Amazon S3 (SSE-S3).

S3 Express One Zone le ofrece la opción de elegir el algoritmo de suma de comprobación que se utiliza para validar los datos durante la carga o descarga. Puede seleccionar uno de los siguientes algoritmos de comprobación de integridad de datos Secure Hash Algoritms (SHA) o Cyclic Redundancy Check (CRC): CRC32, CRC32C, SHA-1 y SHA-256. Las sumas de comprobación basadas en MD5 no son compatibles con la clase de almacenamiento S3 Express One Zone.

Para obtener más información, consulte Prácticas recomendadas adicionales para la suma de comprobación de S3.

AWS Signature Version 4 (SigV4)

S3 Express One Zone utiliza AWS Signature Version 4 (SigV4). SigV4 es un protocolo de firma que se utiliza para autenticar las solicitudes a Amazon S3 a través de HTTPS. S3 Express One Zone firma las solicitudes con AWS Sigv4. Para obtener más información, consulte Autenticación de solicitudes (AWS Signature Version 4) en la Referencia de la API de Amazon Simple Storage Service.

Consistencia sólida

S3 Express One Zone proporciona una sólida coherencia de lectura tras escritura para las solicitudes `PUT` y `DELETE` de objetos en sus buckets de directorio en todas las Regiones de AWS. Para obtener más información, consulte Modelo de consistencia de datos de Amazon S3.

Servicios relacionados

Puede utilizar los siguientes Servicios de AWS con la clase de almacenamiento S3 Express One Zone para admitir su caso de uso específico de baja latencia.

- Amazon Elastic Compute Cloud (Amazon EC2): Amazon EC2 proporciona capacidad de computación escalable y segura en la Nube de AWS. El uso de Amazon EC2 reduce la necesidad de invertir inicialmente en hardware, de manera que puede desarrollar e implementar aplicaciones en menos tiempo. Puede usar Amazon EC2 para lanzar tantos servidores virtuales como necesite, configurar la seguridad y las redes, y administrar el almacenamiento.
- AWS Lambda: Lambda es un servicio informático que permite ejecutar código sin aprovisionar ni administrar servidores. Puede configurar las opciones de notificación en un bucket y conceder a Amazon S3 permiso para invocar una función en la política de permisos basada en recursos de la función.
- Amazon Elastic Kubernetes Service (Amazon EKS): Amazon EKS es un servicio administrado que elimina la necesidad de instalar, operar y mantener su propio plano de control de Kubernetes en AWS. Kubernetes es un sistema de código abierto que automatiza la administración, el escalado y la implementación de aplicaciones en contenedores.
- Amazon Elastic Container Service (Amazon ECS): Amazon ECS es un servicio de orquestación de contenedores completamente administrado que facilita la implementación, la administración y el escalado de aplicaciones en contenedores.
- Amazon Athena: Athena es un servicio de consultas interactivo que facilita el análisis de datos directamente en Amazon S3 con SQL estándar. También puede usar Athena para ejecutar análisis de datos de forma interactiva mediante Apache Spark sin tener que planificar, configurar ni administrar los recursos. Cuando ejecuta aplicaciones de Apache Spark en Athena, envía el código de Spark para su procesamiento y recibe los resultados directamente.
- Entrenamiento del modelo de tiempo de ejecución de Amazon SageMaker: el tiempo de ejecución de Amazon SageMaker es un servicio de machine learning completamente administrado. El tiempo de ejecución de SageMaker permite a los desarrolladores y a los analistas de datos crear y entrenar modelos de machine learning de forma rápida y sencilla y, a continuación, implementarlos directamente en un entorno alojado listo para producción.
- AWS Glue: AWS Glue es un servicio de integración de datos sin servidor que facilita a los usuarios de análisis descubrir, preparar, trasladar e integrar datos desde varios orígenes. Puede usar AWS Glue para análisis, machine learning y desarrollo de aplicaciones. AWS Glue también incluye herramientas adicionales de productividad y operaciones de datos para la creación, la ejecución de trabajos y la implementación de flujos de trabajo empresariales.

- Amazon EMR: Amazon EMR es una plataforma de clúster administrada que simplifica la ejecución de los marcos de macrodatos, tales como Apache Hadoop y Apache Spark, en AWS para procesar y analizar grandes cantidades de datos.

Siguientes pasos

Para obtener más información acerca de cómo trabajar con la clase de almacenamiento S3 Express One Zone y los buckets de directorio, consulte los siguientes temas:

- ¿En qué se diferencia S3 Express One Zone?
- Tutorial: introducción a S3 Express One Zone
- Redes para S3 Express One Zone
- Buckets de directorio
- Uso de objetos en un bucket de directorio
- Seguridad para S3 Express One Zone
- Optimización del rendimiento de Amazon S3 Express One Zone
- Desarrollo con S3 Express One Zone

¿En qué se diferencia S3 Express One Zone?

Amazon S3 Express One Zone es una clase de almacenamiento de Amazon S3 en zona única de alto rendimiento que está diseñada específicamente para ofrecer acceso constante a los datos en milisegundos de un solo dígito para los datos a los que accede para las aplicaciones sensibles a la latencia. S3 Express One Zone es la primera clase de almacenamiento de S3 en la que se puede seleccionar una única zona de disponibilidad con la opción de coubicar su almacenamiento de objetos junto con sus recursos informáticos, lo que brinda la mayor velocidad de acceso posible. Además, para aumentar aún más la velocidad de acceso y admitir cientos de miles de solicitudes por segundo, los datos de S3 Express One Zone se almacenan en un nuevo tipo de bucket: un bucket de directorio de Amazon S3.

Para obtener más información, consulte ¿Qué es S3 Express One Zone? y Buckets de directorio.

Puede crear buckets de directorio y acceder a sus datos en S3 Express One Zone mediante la API de Amazon S3. La API de Amazon S3 es compatible con S3 Express One Zone y los buckets de directorio, con la excepción de algunas diferencias notables. Para obtener más información acerca de cómo se diferencia S3 Express One Zone, consulte los siguientes temas.

Temas

Diferencias de S3 Express One Zone

- Tipo de bucket compatible: los objetos de la clase de almacenamiento S3 Express One Zone solo se pueden almacenar en buckets de directorio. Para obtener más información, consulte Buckets de directorio.
- Durabilidad: con S3 Express One Zone, sus datos se almacenan de forma redundante en varios dispositivos dentro de una única zona de disponibilidad. S3 Express One Zone está diseñado para ofrecer una disponibilidad del 99,95 % dentro de una única zona de disponibilidad y está respaldado por el acuerdo de nivel de servicios de Amazon S3. Para obtener más información, consulte Una sola zona de disponibilidad.
- Comportamiento de **`ListObjectsV2`**
  - Para buckets de directorio, `ListObjectsV2` no devuelve objetos en orden lexicográfico (alfabético). Además, los prefijos deben terminar en un delimitador y solo se puede especificar "/" como delimitador.
  - Para buckets de directorio, la respuesta `ListObjectsV2` incluye los prefijos que están relacionados solo con las cargas multiparte en curso.
- Comportamiento de eliminación: al eliminar un objeto de un bucket de directorio, Amazon S3 elimina de forma recursiva todos los directorios vacíos de la ruta del objeto. Por ejemplo, si elimina la clave del objeto `dir1/dir2/file1.txt`, Amazon S3 elimina `file1.txt`. Si los directorios `dir1/` y `dir2/` están vacíos y no contienen otros objetos, Amazon S3 también los elimina.
- ETags y sumas de comprobación: las etiquetas de entidad (ETag) de S3 Express One Zone son cadenas alfanuméricas aleatorias y no sumas de comprobación MD5. Para obtener más información sobre el uso de sumas de comprobación adicionales con S3 Express One Zone, consulte Prácticas recomendadas adicionales para la suma de comprobación de S3.
- Claves de objetos en solicitudes **`DeleteObjects`**
  - Las claves de objetos de las solicitudes `DeleteObjects` deben contener al menos un carácter que no sea un espacio en blanco. No se admiten cadenas que contengan solo espacios en blanco en solicitudes `DeleteObjects`.

  - Las claves de objeto en solicitudes `DeleteObjects` no pueden contener caracteres de control de Unicode, excepto los caracteres de nueva línea (`\n`), tabulador (`\t`) y retorno de carro (`\r`).
- Puntos de conexión regionales y zonales: al utilizar S3 Express One Zone, debe especificar la región en todas las solicitudes de los clientes. En el caso de los puntos de conexión regionales, debe especificar la región, por ejemplo, `s3express-control.us-west-2.amazonaws.com`. En el caso de los puntos de conexión zonales, se especifica tanto la región como la zona de disponibilidad, por ejemplo, `s3express-usw2-az1.us-west-2.amazonaws.com`. Para obtener más información, consulte Puntos de conexión regionales y zonales.
- Cargas multiparte: al igual que con otros objetos almacenados en Amazon S3, puede cargar y copiar objetos grandes que estén almacenados en la clase de almacenamiento S3 Express One Zone mediante el proceso de carga multiparte. Sin embargo, a continuación se indican algunas diferencias al utilizar el proceso de carga multiparte con objetos almacenados en S3 Express One Zone. Para obtener más información, consulte the section called “Uso de las cargas multiparte con buckets de directorio”.
  - La fecha de creación del objeto es la fecha de finalización de la carga multiparte.
  - Los números de partes multiparte deben utilizar números de partes consecutivos. Si intenta completar una solicitud de carga multiparte con números de parte no consecutivos, Amazon S3 genera un error `400 (Bad Request)` de HTTP.
  - El iniciador de una carga multiparte solo puede anular la solicitud de carga multiparte si se le ha concedido acceso explícito a `AbortMultipartUpload` mediante el permiso `s3express:CreateSession`. Para obtener más información, consulte AWS Identity and Access Management (IAM) para S3 Express One Zone.
- Vaciado de un bucket de directorio: el comando `s3 rm` mediante la AWS Command Line Interface (CLI), la operación `delete` mediante Mountpoint y el botón de opciones del bucket Vacío mediante la AWS Management Console no pueden eliminar las cargas multiparte en curso en un bucket de directorio. Para eliminar estas cargas multiparte en curso, utilice la operación `ListMultipartUploads` para mostrar las cargas multiparte en curso en el bucket y utilice la operación `AbortMultupartUpload` para anular todas las cargas multiparte en curso.

Operaciones de la API compatibles con S3 Express One Zone

La clase de almacenamiento Amazon S3 Express One Zone admite operaciones de la API de puntos de conexión regionales (nivel de bucket o plano de control) y zonales (nivel de objeto o plano de datos). Para obtener más información, consulte Redes para S3 Express One Zone y Puntos de conexión y puntos de conexión de VPC de puerta de enlace.

Operaciones de la API de puntos de conexión regionales

S3 Express One Zone admite las siguientes operaciones de la API de puntos de conexión regionales:

- CreateBucket
- DeleteBucket
- DeleteBucketPolicy
- GetBucketPolicy
- ListDirectoryBuckets
- PutBucketPolicy

Operaciones de la API de puntos de conexión zonales

S3 Express One Zone admite las siguientes operaciones de la API de punto de conexión zonales:

- CreateSession
- CopyObject
- DeleteObject
- DeleteObjects
- GetObject
- GetObjectAttributes
- HeadBucket
- HeadObject
- ListObjectsV2
- PutObject
- AbortMultipartUpload
- CompleteMultiPartUpload
- CreateMultipartUpload
- ListMultipartUploads
- ListParts
- UploadPart
- UploadPartCopy

Características de Amazon S3 no compatibles con S3 Express One Zone

Las siguientes características de Amazon S3 no son compatibles con S3 Express One Zone:

- Políticas administradas de AWS
- AWS PrivateLink para S3
- Sumas de comprobación MD5
- Eliminación de la autenticación multifactor (MFA)
- Bloqueo de objetos de S3
- Pago por el solicitante
- Permisos de acceso de S3
- Puntos de acceso de S3
- Etiquetas de bucket
- Métricas de solicitud de Amazon CloudWatch
- Notificaciones de eventos de S3
- Ciclo de vida de S3
- Puntos de acceso de varias regiones de S3
- Puntos de acceso de S3 Object Lambda
- Control de versiones de S3
- Inventario de S3
- Replicación de S3
- Etiquetas de objetos
- S3 Select
- Registros de acceso al servidor
- Alojamiento de sitios web estáticos
- Almacenamiento de lente de S3
- Grupos de S3 Storage Lens
- S3 Transfer Acceleration
- Cifrado del servidor con claves de AWS Key Management Service (AWS KMS) (SSE-KMS)
- Cifrado del lado del servidor con claves AWS Key Management Service (AWS KMS) (SSE-KMS)

- Cifrado en el servidor con claves proporcionadas por el cliente (SSE-C)
- La opción de copiar la configuración de un bucket existente al crear un bucket nuevo en AWS Management Console.

Tutorial: introducción a S3 Express One Zone

Amazon S3 Express One Zone es la primera clase de almacenamiento de S3 en la que se puede seleccionar una única zona de disponibilidad con la opción de coubicar su almacenamiento de objetos junto con sus recursos de computación, lo que ofrece la mayor velocidad de acceso posible. Los datos de S3 Express One Zone se almacenan en buckets de directorio de S3. Para obtener más información acerca de los buckets de directorio, consulte Buckets de directorio.

S3 Express One Zone es ideal para cualquier aplicación en la que sea importante minimizar la latencia de las solicitudes. Estas aplicaciones pueden ser flujos de trabajo interactivos entre humanos, como la edición de vídeo, en los que los profesionales creativos necesitan un acceso fluido al contenido desde sus interfaces de usuario. S3 Express One Zone también beneficia a las cargas de trabajo de análisis y machine learning que tienen requisitos de capacidad de respuesta similares a los de sus datos, especialmente las cargas de trabajo con muchos accesos más pequeños o un gran número de accesos aleatorios. S3 Express One Zone puede utilizarse con otros servicios de AWS como Amazon EMR, Amazon Athena, AWS Glue Data Catalog y Amazon SageMaker Model Training para admitir cargas de trabajo de análisis, inteligencia artificial y machine learning (IA/ML). Puede trabajar con la clase de almacenamiento S3 Express One Zone y los buckets de directorio mediante la consola de Amazon S3, los AWS SDK, la interfaz de la línea de comandos de AWS (CLI de AWS) y la API de REST de Amazon S3. Para obtener más información, consulte ¿Qué es S3 Express One Zone? y ¿En qué se diferencia S3 Express One Zone?
Este es un diagrama de flujo de trabajo de S3 Express One Zone.

Objetivo

En este tutorial, aprenderá a crear un punto de conexión de puerta de enlace, a crear y asociar una política de IAM, a crear un bucket de directorio y, a continuación, a utilizar la acción de importación para rellenar el bucket de directorio con los objetos que actualmente están almacenados en el bucket de uso general. También puede cargar objetos de forma manual en el bucket de directorio.

Temas

- Requisitos previos
- Paso 1: configure un punto de conexión de VPC de puerta de enlace

Requisitos previos

Antes de empezar este tutorial, debe tener una Cuenta de AWS en la que puede iniciar sesión como usuario de AWS Identity and Access Management (IAM) con los permisos correctos.

Pasos secundarios

- Crear un Cuenta de AWS
- Creación de un usuario de IAM en su Cuenta de AWS (consola)
- Creación de una política de IAM personalizada y asociación de esta a un rol o usuario de IAM (consola)

Crear un Cuenta de AWS

Para completar este tutorial, se necesita una Cuenta de AWS. Cuando se registra en AWS, su Cuenta de AWS se registra automáticamente en todos los servicios de AWS, incluido Amazon S3. Solo se le cobrará por los servicios que utilice. Para obtener más información acerca de los precios, consulte Precios de S3.

Creación de un usuario de IAM en su Cuenta de AWS (consola)

AWS Identity and Access Management (IAM) es un Servicio de AWS que ayuda a los administradores a controlar de forma segura el acceso a los recursos de AWS. Los administradores de IAM controlan quién puede autenticarse (iniciar sesión) y autorizarse (tener permisos) para acceder a los objetos y utilizar buckets de directorio en S3 Express One Zone. El uso de IAM no está sujeto a ningún cargo adicional.

De forma predeterminada, los usuarios no tienen permisos para acceder a buckets de directorio y realizar operaciones en S3 Express One Zone. Para conceder permisos de acceso a los buckets

de directorio y a las operaciones de S3 Express One Zone, puede usar IAM para crear usuarios o roles y asociar permisos a esas identidades. Para obtener más información acerca de cómo crear un usuario de IAM, consulte Creación del primer grupo de usuarios y administradores de IAM en la guía del usuario de IAM. Para obtener más información sobre cómo crear un rol de IAM, consulte Creación de un rol para delegar permisos a un usuario de IAM en la Guía del usuario de IAM.

Para simplificar, este tutorial crea y utiliza un usuario de IAM. Después de completar este tutorial, recuerde Eliminación del rol de IAM. Para uso en producción, le recomendamos que siga las Prácticas recomendadas de seguridad de IAM en la Guía del usuario de IAM. Una práctica recomendada exige que los usuarios humanos utilicen la federación con un proveedor de identidades para acceder a AWS con credenciales temporales. Otra práctica recomendada es exigir a las cargas de trabajo que utilicen credenciales temporales con roles de IAM para acceder a AWS. Para obtener más información sobre el uso de AWS IAM Identity Center para crear usuarios con credenciales temporales, consulte Introducción en la Guía del usuario de AWS IAM Identity Center.

> **Warning**
>
> Los usuarios de IAM tienen credenciales de larga duración, lo que supone un riesgo de seguridad. Para ayudar a mitigar este riesgo, le recomendamos que brinde a estos usuarios únicamente los permisos que necesitan para realizar la tarea y que los elimine cuando ya no los necesiten.

Creación de una política de IAM personalizada y asociación de esta a un rol o usuario de IAM (consola)

De forma predeterminada, los usuarios no tienen permisos para los buckets de directorio ni para las operaciones de S3 Express One Zone. Para conceder permisos de acceso a los buckets de directorio, puede usar IAM para crear usuarios, grupos o roles y asociar permisos a esas identidades. Los buckets de directorio son el único recurso que puede incluir en las políticas de bucket o en las políticas de identidad de IAM para el acceso a S3 Express One Zone.

Para utilizar las operaciones de la API de puntos de conexión regionales (operaciones de bucket o plano de control) con S3 Express One Zone, utilice el modelo de autorización de IAM, que no implica la administración de sesiones. Los permisos se conceden para las acciones de forma individual. Para utilizar las operaciones de la API de puntos de conexión zonales (operaciones de objeto o plano de datos), utilice `CreateSession` para crear y administrar sesiones optimizadas para la autorización de solicitudes de datos con baja latencia. Para recuperar y usar un token de sesión, debe permitir

la acción `s3express:CreateSession` para su bucket de directorio en una política basada en identidades o en una política de bucket. Si accede a S3 Express One Zone en la consola de Amazon S3, a través de la interfaz de la línea de comandos de AWS (CLI de AWS) o mediante los AWS SDK, S3 Express One Zone crea una sesión en su nombre. Para obtener más información, consulte Autorización de `CreateSession` y AWS Identity and Access Management (IAM) para S3 Express One Zone.

Creación de una política de IAM y asociación de esta a un usuario (o rol) de IAM

1. Inicie sesión en la consola de administración de AWS y abra la consola de administración de IAM.
2. En el panel de navegación, seleccione Políticas.
3. Seleccione Crear política.
4. Seleccione JSON.
5. Copie la siguiente política en la ventana del Editor de políticas. Para poder crear buckets de directorio o utilizar S3 Express One Zone, debe conceder los permisos necesarios a su rol o usuarios de AWS Identity and Access Management (IAM). Esta política de ejemplo permite el acceso a la operación de la API `CreateSession` (para utilizarla con las operaciones de la API zonales o de objeto) y a todas las operaciones de la API del punto de conexión regional (de bucket). Esta política permite que la operación de la API `CreateSession` se utilice con todos los buckets de directorio, pero las operaciones de la API de punto de conexión regional solo se permiten con el bucket de directorio especificado. Para utilizar esta política de ejemplo, sustituya *`user input placeholders`* por su propia información.

```
{
     "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": "*"
        }
    ]
}
```

6. Elija Siguiente.
7. Asigne un nombre a la política.

> **Note**
>
> Las etiquetas de bucket no son compatibles con S3 Express One Zone.

8. Seleccione Crear política.
9. Ahora que ha creado una política de IAM, puede asociarla a un usuario de IAM. En el panel de navegación, seleccione Políticas.
10. En la barra de búsqueda, escriba el nombre de la política.
11. En el menú Acciones, seleccione Asociar.
12. En Filtrar por tipo de entidad, seleccione Usuarios de IAM o Roles.
13. En el campo de búsqueda, escriba el nombre del usuario o rol que desee utilizar.
14. Seleccione Asociar política.

Paso 1: configure un punto de conexión de VPC de puerta de enlace

Puede acceder a las operaciones de API zonales y regionales a través de puntos de conexión de la nube privada virtual (VPC) de puerta de enlace. Los puntos de conexión de puerta de enlace pueden permitir que el tráfico llegue a S3 Express One Zone sin atravesar una puerta de enlace NAT. Recomendamos encarecidamente utilizar puntos de conexión de puerta de enlace, ya que son la ruta de red más óptima cuando se trabaja con S3 Express One Zone. Puede acceder a buckets de directorio de S3 Express One Zone desde su VPC sin necesidad de una puerta de enlace de Internet ni de un dispositivo NAT para la VPC, y sin costo adicional. Utilice el siguiente procedimiento para

configurar un punto de conexión de una puerta de enlace que se conecte a buckets de directorio y objetos de clase de almacenamiento de S3 Express One Zone.

Para acceder a S3 Express One Zone, utilice puntos de conexión regionales y zonales que sean diferentes de los puntos de conexión estándar de Amazon S3. Según la operación de API de Amazon S3 que utilice, es necesario un punto de conexión zonal o regional. Para obtener una lista completa de las operaciones de API admitidas por tipo de punto de conexión, consulte Operaciones de la API compatibles con S3 Express One Zone. Debe acceder a los puntos de conexión zonales y regionales a través de un punto de conexión de la nube privada virtual (VPC) de puerta de enlace.

Utilice el siguiente procedimiento para crear un punto de conexión de una puerta de enlace que se conecte a buckets de directorio y objetos de clase de almacenamiento de S3 Express One Zone.

Para configurar un punto de conexión de VPC de puerta de enlace

1. Abra la consola de Amazon VPC en https://console.aws.amazon.com/vpc/.
2. En el panel de navegación lateral, en Nube virtual privada, seleccione Puntos de conexión.
3. Seleccione Crear punto de conexión.
4. Cree un nombre para el punto de conexión.
5. En Categoría de servicios, elija Servicios de AWS.
6. En Servicios, realice una búsqueda con el filtro Type=Gateway y luego seleccione el botón de opción situado junto a com.amazonaws.***region***.s3express.
7. En VPC, elija la VPC en la que desea crear el punto de conexión.
8. En Tablas de enrutamiento, seleccione las tablas de enrutamiento que debe utilizar el punto de conexión. Amazon VPC agregará automáticamente una ruta para dirigir el tráfico destinado al servicio a la interfaz de red del punto de conexión.
9. En Política, elija Acceso completo para permitir todas las operaciones de todas las entidades principales en todos los recursos del punto de conexión de VPC. De lo contrario, elija Personalizar para adjuntar una política de punto de conexión de VPC que controle los permisos que tienen las entidades principales para realizar acciones en los recursos a través del punto de conexión de VPC.
10. Seleccione Crear punto de conexión.

Tras crear un punto de conexión de puerta de enlace, puede utilizar los puntos de conexión de API regionales y los puntos de conexión de API zonales para acceder a buckets de directorio y objetos de clase de almacenamiento Amazon S3 Express One Zone.

Paso 2: cree un bucket de directorio

1. Inicie sesión AWS Management Console Management Console y abra la consola de Amazon S3 en https://console.aws.amazon.com/s3/.
2. En la barra de navegación de la parte superior de la página, elija el nombre de la Región de AWS que aparece. A continuación, elija en Region (Región) la región en la que desea crear un bucket.

   > **Note**
   >
   > Puede seleccionar una región cercana para minimizar la latencia y los costos, así como para satisfacer los requisitos normativos. Los objetos almacenados en una región nunca abandonarán esa región salvo que usted los transfiera de forma específica a otra. Para una lista de Regiones de AWS de Amazon S3, consulte Puntos de conexión de Servicio de AWS en la Referencia general de Amazon Web Services.

3. En el panel de navegación izquierdo, elija Instancias.
4. Elija Crear bucket.

   Se abrirá la página Crear bucket.
5. En Configuración general, vea la Región de AWS donde se creará el bucket.
6. En Bucket type (Tipo de bucket), elija Directory (Directorio).

   > **Note**
   >
   > - Si ha elegido una región que no admite buckets de directorio, la opción Tipo de bucket desaparece y el tipo de bucket pasa a ser un bucket de uso general de forma predeterminada. Para crear un bucket de directorio, debe elegir una región compatible. Para obtener una lista de las regiones que admiten buckets de directorio y la clase de almacenamiento Amazon S3 Express One Zone, consulte the section called “Zonas y regiones de disponibilidad de S3 Express One Zone”.
   > - No se puede cambiar el tipo de bucket después de haberlo creado.

   En Zona de disponibilidad, elija una zona de disponibilidad local para sus servicios de computación. Para obtener una lista de las zonas de disponibilidad que admiten buckets de

directorio y la clase de almacenamiento S3 Express One Zone, consulte the section called "Zonas y regiones de disponibilidad de S3 Express One Zone".

> **Note**
>
> La zona de disponibilidad no se puede cambiar una vez creado el bucket.

7. En Zona de disponibilidad, selecciona la casilla para confirmar que, en caso de que se produzca una interrupción en la zona de disponibilidad, es posible que sus datos no estén disponibles o se pierdan.

> **Important**
>
> Los buckets de directorio se almacenan en varios dispositivos dentro de una única zona de disponibilidad, pero no almacenan datos de forma redundante en todas las zonas de disponibilidad.

8. En Nombre del bucket, escriba un nombre para el bucket de directorio.

   Las siguientes reglas de nomenclatura se aplican a los buckets de directorio.

   - Ser únicos dentro de la Región de AWS y la zona de disponibilidad elegida.
   - El nombre debe tener entre 3 (mín.) y 63 caracteres (máx.), incluido el sufijo.
   - Constar de letras minúsculas, números y guiones (-).
   - Comenzar y terminar por un número o una letra.
   - Debe incluir el siguiente sufijo: `--`*`azid`*`--x-s3`.
   - Los nombres de los buckets no deben comenzar con el prefijo `xn--`.
   - Los nombres de los buckets no deben comenzar con el prefijo `sthree-`.
   - Los nombres de los buckets no deben comenzar con el prefijo `sthree-configurator`.
   - Los nombres de los buckets no deben comenzar con el prefijo `amzn-s3-demo-`.
   - Los nombres de los buckets no deben terminar con el sufijo `-s3alias`. Este sufijo está reservado para nombres de alias de punto de acceso. Para obtener más información, consulte Uso de un alias de estilo de bucket para su punto de acceso de bucket de S3.
   - Los nombres de los buckets no deben terminar con el sufijo `--ol-s3`. Este sufijo está reservado para nombres de alias de punto de acceso de Object Lambda. Para obtener más

información, consulte Cómo usar un alias de estilo de bucket para su punto de acceso de Object Lambda de bucket de S3.

- Los nombres de los buckets no deben terminar con el sufijo `.mrap`. Este sufijo está reservado para nombres de punto de acceso de varias regiones. Para obtener más información, consulte Reglas para asignar nombres a los puntos de acceso de varias regiones de Amazon S3.

Se añade automáticamente un sufijo al nombre base que proporciona cuando crea un bucket de directorio utilizando la consola. Este sufijo incluye el ID de zona de disponibilidad de la que haya elegido.

Una vez que haya creado el bucket, no podrá modificar su nombre. Para obtener más información sobre la nomenclatura de los buckets, consulte Reglas de nomenclatura de buckets.

> **Important**
>
> No incluya información confidencial, como números de cuenta, en el nombre del bucket. El nombre del bucket será visible en las URL que señalan a los objetos almacenados en él.

9. En Propiedad del objeto, la opción Aplicada al propietario del bucket se activa de forma automática y se desactivan todas las listas de control de acceso (ACL). En el caso de los buckets de directorio, las ACL no se pueden habilitar.

   ACL desactivadas

   - Propietario del bucket obligatorio (predeterminado): las ACL están desactivadas y el propietario del bucket tiene automáticamente la propiedad y el control total sobre cada objeto del bucket. Las ACL ya no afectan a los permisos de acceso de los datos del bucket de S3. El bucket utiliza políticas exclusivamente para definir el control de acceso.

     La mayoría de los casos de uso modernos de Amazon S3 ya no requieren el uso de ACL. Para obtener más información, consulte Control de la propiedad de los objetos y desactivación de las ACL del bucket.

10. En Configuración de bloqueo de acceso público para este bucket, tenga en cuenta que la configuración Bloquear acceso público para su bucket de directorio se habilita automáticamente. Esta configuración no se puede modificar para los bucket de directorio. Para obtener más información acerca del bloqueo del acceso público, consulte Bloquear el acceso público a su almacenamiento de Amazon S3.

11. En Configuración del cifrado del lado del servidor, Amazon S3 aplica el cifrado del lado del servidor con claves administradas por Amazon S3 (SSE-S3) como nivel básico de cifrado para todos los buckets de S3. Todas las cargas de objetos en buckets de directorio se cifran con SSE-S3. No se puede modificar el tipo de cifrado para los bucket de directorio. Para obtener más información sobre SSE-S3, consulte the section called “Claves de cifrado administradas por Amazon S3 (SSE-S3)”.
12. Elija Crear bucket.

    Después de crear el bucket, puede añadir archivos y carpetas al bucket. Para obtener más información, consulte the section called “Trabajar con objetos en un bucket de directorio”.

En el paso siguiente se muestra cómo utilizar la acción de importación de la consola de Amazon S3 para rellenar el bucket de directorio con datos.

Paso 3: importación de datos a un bucket de directorio

Para completar este paso, debe tener un bucket de uso general que contenga objetos y que esté ubicado en la misma Región de AWS que su bucket de directorio.

Tras crear un bucket de directorio en Amazon S3, puede rellenar el nuevo bucket con datos mediante la acción de importación de la consola de Amazon S3. La importación simplifica la copia de datos en los buckets de directorio, ya que permite elegir un prefijo o un bucket de uso general desde el que importar los datos sin tener que especificar todos los objetos que se van a copiar de forma individual. La importación utiliza Operaciones por lotes de Amazon S3, que copia los objetos en el prefijo seleccionado o en el bucket de uso general. Puede supervisar el progreso del trabajo de importación de copias a través de la página de detalles del trabajo de Operaciones por lotes de Amazon S3.

Uso de la acción de importación

1. Inicie sesión AWS Management Console Management Console y abra la consola de Amazon S3 en https://console.aws.amazon.com/s3/.
2. En la barra de navegación de la parte superior de la página, elija el nombre de la Región de AWS que aparece. A continuación, elija la región asociada a la zona de disponibilidad en la que se encuentra su bucket de directorio.
3. En el panel de navegación de la izquierdo, seleccione Buckets y, a continuación, la pestaña Buckets de directorio. Seleccione el bucket de directorio al que desea importar objetos.
4. Seleccione Importar.

5. En Origen, introduzca el bucket de uso general (o la ruta del bucket, incluido el prefijo) que contiene los objetos que desea importar. Para elegir un bucket de uso general existente de una lista, seleccione Examinar S3.
6. En la sección Permisos, puede elegir que un rol de IAM se genere automáticamente. También puede seleccionar un rol de IAM de una lista o introducir directamente un ARN de rol de IAM.

   - Para que Amazon S3 pueda crear un nuevo rol de IAM en su nombre, seleccione Crear un nuevo rol de IAM.

     > **Note**
     >
     > Si los objetos de origen se cifran en el servidor con claves de AWS Key Management Service (AWS KMS) (SSE-KMS), no elija la opción Crear un nuevo rol de IAM. En su lugar, especifique un rol de IAM existente que tenga el permiso `kms:Decrypt`. Amazon S3 utilizará este permiso para descifrar sus objetos. Durante el proceso de importación, Amazon S3 volverá a cifrar esos objetos mediante un cifrado del servidor con claves administradas de Amazon S3 (SSE-S3).

   - Para elegir un rol de IAM existente de una lista, seleccione Elegir entre los roles de IAM existentes.
   - Para especificar un rol de IAM existente con su nombre de recurso de Amazon (ARN), seleccione Introducir ARN del rol de IAM y, a continuación, introduzca el ARN en el campo correspondiente.
7. Revise la información que aparece en las secciones Destino y Configuración de objetos copiada. Si la información de la sección Destino es correcta, seleccione Importar para iniciar el trabajo de copia.

   La consola de Amazon S3 muestra el estado del nuevo trabajo en la página Operaciones por lotes. Para obtener más información sobre el trabajo, pulse el botón de opción junto al nombre del trabajo y, a continuación, en el menú Acciones, seleccione Ver detalles. Para abrir el bucket de directorio al que se importarán los objetos, seleccione Ver destino de la importación.

Paso 4: cargue manualmente los objetos a su bucket de directorio

También puede cargar objetos de forma manual en el bucket de directorio.

Carga manual de objetos

1. Inicie sesión AWS Management Console Management Console y abra la consola de Amazon S3 en https://console.aws.amazon.com/s3/.
2. En la barra de navegación de la esquina superior derecha de la página, elija el nombre de la Región de AWS que se muestra actualmente. A continuación, elija la región asociada a la zona de disponibilidad en la que se encuentra su bucket de directorio.
3. En el panel de navegación situado a la izquierda, elija Buckets.
4. Seleccione la pestaña Buckets de directorio.
5. Elija el nombre del bucket en el que desea cargar sus carpetas o archivos.

   > **Note**
   >
   > Si eligió el mismo bucket de directorio que utilizó en los pasos anteriores de este tutorial, el bucket de directorio contendrá los objetos que se cargaron desde la herramienta de importación. Observe que estos objetos ahora están almacenados en la clase de almacenamiento S3 Express One Zone.

6. En la pestaña Objetos, elija Cargar.
7. En la página Cargar, lleve a cabo alguna de las siguientes acciones:

   - Arrastre y suelte archivos y carpetas en el área de carga punteada.
   - Elija Agregar archivos o Agregar carpeta, elija los archivos o carpetas que desee cargar y, a continuación, seleccione Abrir o Cargar.

8. En Sumas de comprobación, seleccione la función de suma de comprobación que desee usar.

   > **Note**
   >
   > Recomendamos utilizar CRC32 y CRC32C para obtener el mejor rendimiento con la clase de almacenamiento S3 Express One Zone. Para obtener más información, consulte Prácticas recomendadas adicionales para la suma de comprobación de S3.

   (Opcional) Si va a cargar un único objeto de un tamaño inferior a 16 MB, también puede especificar un valor de suma de comprobación precalculado. Al proporcionar un valor precalculado, Amazon S3 lo compara con el valor que calcula mediante la función de suma de comprobación seleccionada. Si los valores no coinciden, la carga no se iniciará.

9. Las opciones de las secciones Permisos y Propiedades se configuran automáticamente con la configuración predeterminada y no se pueden modificar. El bloqueo de acceso público se habilita automáticamente. El control de versiones de S3 y el bloqueo de objetos de S3 no se pueden habilitar en los buckets de directorio.

   (Opcional) Si quiere añadir metadatos en pares clave-valor a sus objetos, expanda la sección Propiedades y, a continuación, en la sección Metadatos, elija Agregar metadatos.
10. Para cargar inmediatamente los archivos y carpetas indicados, elija Cargar.

    Amazon S3 carga sus objetos y carpetas. Cuando finalice la carga, puede ver un mensaje de éxito en la página Cargar: estado.

    Ha creado correctamente un bucket de directorio y ha cargado objetos en su bucket.

Paso 5: vacíe el bucket de directorio

Puede vaciar su bucket de directorio de Amazon S3 con la consola de Amazon S3.

Cómo vaciar un bucket de directorio

1. Inicie sesión AWS Management Console Management Console y abra la consola de Amazon S3 en https://console.aws.amazon.com/s3/.
2. En la barra de navegación de la esquina superior derecha de la página, elija el nombre de la Región de AWS que se muestra actualmente. A continuación, elija la región asociada a la zona de disponibilidad en la que se encuentra su bucket de directorio.
3. En el panel de navegación situado a la izquierda, elija Buckets.
4. Seleccione la pestaña Buckets de directorio.
5. Elija el botón de opción junto al nombre del bucket que desea vaciar y, a continuación, Vaciar.
6. En la página Empty bucket (Vaciar bucket), confirme que desea vaciar el bucket; para ello, ingrese **`permanently delete`** en el campo de texto y luego, elija Empty (Vaciar).
7. Monitoree el progreso del proceso de vaciado del bucket en la página Vaciar bucket: estado.

Paso 6: elimine su bucket de directorio

Después de vaciar el bucket de directorio y anular todas las cargas multiparte en curso, ya puede eliminar el bucket utilizando la consola de Amazon S3.

Eliminación de un bucket de directorio

1. Inicie sesión AWS Management Console Management Console y abra la consola de Amazon S3 en https://console.aws.amazon.com/s3/.
2. En la barra de navegación de la esquina superior derecha de la página, elija el nombre de la Región de AWS que se muestra actualmente. A continuación, elija la región asociada a la zona de disponibilidad en la que se encuentra su bucket de directorio.
3. En el panel de navegación situado a la izquierda, elija Buckets.
4. Seleccione la pestaña Buckets de directorio.
5. En la lista Buckets de directorio, elija el botón de opción junto al bucket que desea eliminar.
6. Elija Eliminar.
7. En la página Eliminar bucket, escriba el nombre del bucket en el campo de texto para confirmar la eliminación del bucket.

   > **Important**
   >
   > La eliminación de un bucket de directorio no se puede revertir.

8. Para eliminar el bucket de directorio, elija Eliminar bucket.

Siguientes pasos

En este tutorial, ha aprendido a crear un bucket de directorio y a utilizar la clase de almacenamiento S3 Express One Zone. Después de completar este tutorial, puede explorar los servicios relacionados de AWS que puede utilizar con la clase de almacenamiento S3 Express One Zone.

Puede utilizar los siguientes Servicios de AWS con la clase de almacenamiento S3 Express One Zone para admitir su caso de uso específico de baja latencia.

- Amazon Elastic Compute Cloud (Amazon EC2): Amazon EC2 proporciona capacidad de computación escalable y segura en la Nube de AWS. El uso de Amazon EC2 reduce la necesidad de invertir inicialmente en hardware, de manera que puede desarrollar e implementar aplicaciones en menos tiempo. Puede usar Amazon EC2 para lanzar tantos servidores virtuales como necesite, configurar la seguridad y las redes, y administrar el almacenamiento.
- AWS Lambda: Lambda es un servicio informático que permite ejecutar código sin aprovisionar ni administrar servidores. Puede configurar las opciones de notificación en un bucket y conceder a

Amazon S3 permiso para invocar una función en la política de permisos basada en recursos de la función.

- Amazon Elastic Kubernetes Service (Amazon EKS): Amazon EKS es un servicio administrado que elimina la necesidad de instalar, operar y mantener su propio plano de control de Kubernetes en AWS. Kubernetes es un sistema de código abierto que automatiza la administración, el escalado y la implementación de aplicaciones en contenedores.
- Amazon Elastic Container Service (Amazon ECS): Amazon ECS es un servicio de orquestación de contenedores completamente administrado que facilita la implementación, la administración y el escalado de aplicaciones en contenedores.
- Amazon EMR: Amazon EMR es una plataforma de clúster administrada que simplifica la ejecución de los marcos de macrodatos, tales como Apache Hadoop y Apache Spark, en AWS para procesar y analizar grandes cantidades de datos.
- Amazon Athena: Athena es un servicio de consultas interactivo que facilita el análisis de datos directamente en Amazon S3 con SQL estándar. También puede usar Athena para ejecutar análisis de datos de forma interactiva mediante Apache Spark sin tener que planificar, configurar ni administrar los recursos. Cuando ejecuta aplicaciones de Apache Spark en Athena, envía el código de Spark para su procesamiento y recibe los resultados directamente.
- Catálogo de datos de AWS Glue: AWS Glue es un servicio de integración de datos sin servidor que hace más fácil a los usuarios de análisis descubrir, preparar, trasladar e integrar datos desde varios orígenes. Puede utilizar AWS Glue para análisis, machine learning y desarrollo de aplicaciones. AWS Catálogo de datos de AWS Glue es un repositorio centralizado que almacena metadatos sobre los conjuntos de datos de su organización. Actúa como un índice para las métricas de tiempo de ejecución, esquema y ubicación de sus orígenes de datos.
- Entrenamiento del modelo de tiempo de ejecución de Amazon SageMaker: el tiempo de ejecución de Amazon SageMaker es un servicio de machine learning completamente administrado. El tiempo de ejecución de SageMaker permite a los desarrolladores y a los analistas de datos crear y entrenar modelos de machine learning de forma rápida y sencilla y, a continuación, implementarlos directamente en un entorno alojado listo para producción.

Para obtener más información sobre S3 Express One Zone, consulte ¿Qué es S3 Express One Zone? y ¿En qué se diferencia S3 Express One Zone?

Redes para S3 Express One Zone

Para acceder a buckets de directorio y objetos de clase de almacenamiento Amazon S3 Express One Zone, utilice puntos de conexión de API regionales y zonales que sean diferentes de los puntos de conexión estándar de Amazon S3. Según la operación de API de S3 que utilice, es necesario un punto de conexión zonal o regional. Para obtener una lista completa de las operaciones de API por tipo de punto de conexión, consulte Operaciones de la API compatibles con S3 Express One Zone.

Puede acceder a las operaciones de API zonales y regionales a través de puntos de conexión de la nube privada virtual (VPC) de puerta de enlace. Para configurar los puntos de conexión de VPC de puerta de enlace, consulte the section called “Configuración de puntos de conexión de puerta de enlace de VPC”.

En los siguientes temas se describen los requisitos de red para acceder a S3 Express One Zone mediante un punto de conexión de VPC de puerta de enlace.

Temas

puntos de conexión

Puede acceder a buckets de directorio y objetos de clase de almacenamiento Amazon S3 Express One Zone desde su VPC mediante puntos de conexión de VPC de puerta de enlace. S3 Express One Zone utiliza puntos de conexión de API regionales y zonales. Según la operación de API de Amazon S3 que utilice, es necesario un punto de conexión regional o zonal. El uso de puntos de enlace de gateway no supone ningún cargo adicional.

Las operaciones de API de nivel de bucket (o plano de control) están disponibles a través de los puntos de conexión regionales y se denominan operaciones de la API de puntos de conexión regionales. Algunos ejemplos de operaciones de la API de puntos de conexión regionales son `CreateBucket` y `DeleteBucket`. Al crear un bucket de directorio, elige una única disponibilidad en la que se creará el bucket de directorio. Tras crear un bucket de directorio, puede utilizar las operaciones de la API de puntos de conexión zonales para cargar y administrar los objetos de su bucket de directorio.

Las operaciones de API de nivel de objeto (o plano de datos) están disponibles a través de los puntos de conexión zonales y se denominan operaciones de la API de puntos de conexión zonales.

Algunos ejemplos de operaciones de la API de puntos de conexión zonales son `CreateSession` y `PutObject`.

En la siguiente tabla se muestran los puntos de conexión de las API regionales y zonales que están disponibles para cada región y zona de disponibilidad.

Configuración de puntos de conexión de puerta de enlace de VPC

Utilice el siguiente procedimiento para crear un punto de conexión de una puerta de enlace que se conecte a buckets de directorio y objetos de clase de almacenamiento de Amazon S3 Express One Zone.

Para configurar un punto de conexión de VPC de puerta de enlace

1. Abra la consola de Amazon VPC en https://console.aws.amazon.com/vpc/.
2. En el panel de navegación, elija Puntos de conexión.
3. Elija Crear punto de conexión.
4. Cree un nombre para el punto de conexión.
5. En Categoría de servicios, elija Servicios de AWS.
6. En Servicios, añada el filtro Type=Gateway y luego seleccione el botón de opción situado junto a com.amazonaws.***region***.s3express.
7. En VPC, elija la VPC en la que desea crear el punto de conexión.
8. En Tablas de enrutamiento, seleccione las tablas de enrutamiento que debe utilizar el punto de conexión. Amazon VPC agregará automáticamente una ruta para dirigir el tráfico destinado al servicio a la interfaz de red del punto de conexión.
9. En Política, elija Acceso completo para permitir todas las operaciones de todas las entidades principales en todos los recursos del punto de conexión de VPC. De lo contrario, elija Personalizar para adjuntar una política de punto de conexión de VPC que controle los permisos que tienen las entidades principales para realizar acciones en los recursos a través del punto de conexión de VPC.
10. (Opcional) Para añadir una etiqueta, elija Agregar etiqueta nueva e ingrese la clave y el valor de la etiqueta.
11. Seleccione Crear punto de conexión.

Tras crear un punto de conexión de puerta de enlace, puede utilizar los puntos de conexión de API regionales y los puntos de conexión de API zonales para acceder a buckets de directorio y objetos de clase de almacenamiento Amazon S3 Express One Zone.

Buckets de directorio

Existen dos tipos de buckets de Amazon S3: buckets de uso general y buckets de directorio. Elija el tipo de bucket que mejor se adapte a sus requisitos de rendimiento y aplicación:

- Los bucket de uso general son del tipo de bucket original de S3 y se recomiendan para la mayoría de los casos de uso y patrones de acceso. Los bucket de uso general también permiten almacenar objetos en todas las clases de almacenamiento, excepto en S3 Express One Zone.
- Los buckets de directorio utilizan la clase de almacenamiento S3 Express One Zone, que se recomienda si la aplicación es sensible al rendimiento y se beneficia de latencias `PUT` y `GET` de un solo dígito de milisegundos.

Los buckets de directorio se utilizan para cargas de trabajo o aplicaciones de rendimiento crítico que requieren una latencia constante de milisegundos de un solo dígito. Los buckets de directorio organizan los datos jerárquicamente en directorios, a diferencia de la estructura de almacenamiento plana de los buckets de uso general. No hay límites de prefijos para los buckets de directorio y los directorios individuales pueden realizar un escalado horizontal.

Los buckets de directorio utilizan la clase de almacenamiento S3 Express One Zone, que almacena datos en varios dispositivos dentro de una única zona de disponibilidad, pero no almacenan datos de forma redundante en todas las zonas de disponibilidad Al crear un bucket de directorio, recomendamos que especifique una Región de AWS y una zona de disponibilidad que sean locales para sus instancias de computación de Amazon EC2, Amazon Elastic Kubernetes Service o Amazon Elastic Container Service (Amazon ECS) a fin de optimizar el rendimiento.

Puede crear hasta 10 buckets de directorio en cada una de sus Cuentas de AWS, sin límite en cuanto al número de objetos que puede almacenar en un bucket. La cuota de buckets se aplica a cada región en su Cuenta de AWS. Si su aplicación requiere aumentar este límite, contacte con AWS Support. Para obtener más información, consulte Consola de Service Quotas.

> **Important**
>
> Los buckets de directorio que no tengan actividad de solicitudes durante un periodo de al menos 90 días pasarán a un estado inactivo. Cuando se encuentra en un estado inactivo, un

bucket de directorio queda temporalmente inaccesible para lecturas y escrituras. Los buckets inactivos conservan todo el almacenamiento, los metadatos de objetos y los metadatos del bucket. Los cargos de almacenamiento existentes se aplicarán a los buckets inactivos. Si realiza una solicitud de acceso a un bucket inactivo, el bucket pasará a un estado activo en cuestión de minutos. Durante este periodo de transición, las lecturas y escrituras devolverán un código de error HTTP `503 (Service Unavailable)`.

En los siguientes temas se proporciona información acerca de los buckets de directorio. Para obtener más información acerca de los buckets de uso general, consulte Descripción general de los buckets.

Temas

Zonas de disponibilidad

Al crear un bucket de directorio, elige la zona de disponibilidad y Región de AWS.

Los buckets de directorio utilizan la clase de almacenamiento S3 Express One Zone, que está diseñada para que la utilicen aplicaciones sensibles al rendimiento. S3 Express One Zone es

la primera clase de almacenamiento de S3 en la que se puede seleccionar una única zona de disponibilidad con la opción de coubicar su almacenamiento de objetos junto con sus recursos informáticos, lo que brinda la mayor velocidad de acceso posible.

Con S3 Express One Zone, sus datos se almacenan de forma redundante en varios dispositivos dentro de una única zona de disponibilidad. S3 Express One Zone está diseñado para ofrecer una disponibilidad del 99,95 % dentro de una única zona de disponibilidad y está respaldado por el contrato de nivel de servicio de Amazon S3. Para obtener más información, consulte Una sola zona de disponibilidad

Nombres de los buckets de directorio

El nombre de un bucket de directorio consta de un nombre base que proporciona y un sufijo que contiene el ID de la zona de disponibilidad en la que se encuentra el bucket. Los nombres de los buckets de directorio deben seguir este formato y seguir las reglas de denominación de los buckets de directorio:

```
bucket-base-name--azid--x-s3
```

Por ejemplo, el siguiente nombre del bucket de directorio contiene el ID de zona de disponibilidad `usw2-az1`:

```
bucket-base-name--usw2-az1--x-s3
```

Para obtener más información, consulte Reglas de nomenclatura de buckets de directorio.

Directorios

Los buckets de directorio organizan los datos jerárquicamente en directorios, a diferencia de la estructura de ordenación plana de los buckets de uso general. Cada bucket de directorio de S3 puede admitir cientos de miles de transacciones por segundo (TPS), independientemente del número de directorios dentro del bucket.

Con un espacio de nombres jerárquico, el delimitador de la clave del objeto es importante. El único delimitador admitido es una barra inclinada (`/`). Los directorios se determinan mediante los límites de los delimitadores. Por ejemplo, la clave del objeto `dir1/dir2/file1.txt` hace que los directorios `dir1/` y `dir2/` se creen automáticamente y que el objeto `file1.txt` se añada al directorio `/dir2` de la ruta `dir1/dir2/file1.txt`.

El modelo de indexación de buckets de directorio devuelve resultados sin ordenar de la operación de la API `ListObjectsV2`. Si es necesario limitar los resultados a una subsección del bucket, puede especificar una ruta de subdirectorio en el parámetro `prefix`, por ejemplo, `prefix=dir1/`.

Nombres de claves

En el caso de los buckets de directorio, los subdirectorios comunes a varias claves del objeto se crean con la primera clave del objeto. Las claves de objeto adicionales del mismo subdirectorio utilizan el subdirectorio creado anteriormente. Este modelo ofrece flexibilidad a la hora de elegir las claves del objeto que mejor se adapten a la aplicación, además de admitir directorios dispersos y densos.

Administración de accesos

Los buckets de directorio tienen habilitadas forma predeterminada todas las configuraciones de S3 Block Public Access en el nivel de bucket. S3 Object Ownership está configurada como aplicada al propietario del bucket y las listas de control de acceso (ACL) están deshabilitadas. Esta configuración no se puede modificar.

De forma predeterminada, los usuarios no tienen permisos para los buckets de directorio ni para las operaciones de S3 Express One Zone. Para conceder permisos de acceso a los buckets de directorio, puede usar IAM para crear usuarios, grupos o roles y asociar permisos a esas identidades. Para obtener más información, consulte AWS Identity and Access Management (IAM) para S3 Express One Zone.

Trabajar con buckets de de directorio

Para obtener más información acerca de cómo trabajar con buckets de directorio, consulte los siguientes temas.

Temas

- Reglas de nomenclatura de buckets de directorio
- Crear un bucket de directorio
- Visualización de las propiedades del bucket de directorio
- Administración de políticas de buckets para buckets de directorio
- Vaciado de un bucket de directorio
- Eliminar un bucket de directorio

- Mostrar una lista de buckets de directorio
- Uso de HeadBucket con buckets de directorio

Reglas de nomenclatura de buckets de directorio

Al crear un bucket de directorio en Amazon S3, se aplican las siguientes reglas de nomenclatura del bucket. Para conocer las reglas de nomenclatura de los buckets de uso general, consulte Reglas de nomenclatura de buckets.

El nombre de un bucket de directorio consta del nombre base que proporcione y un sufijo que contiene el ID de la zona de disponibilidad de AWS en la que se encuentra el bucket y `--x-s3`.

```
base-name--azid--x-s3
```

Por ejemplo, el siguiente nombre del bucket de directorio contiene el ID de zona de disponibilidad `usw2-az1`:

```
bucket-base-name--usw2-az1--x-s3
```

> **Note**
>
> Al crear un bucket de directorio mediante la consola, se agrega automáticamente un sufijo al nombre base que proporcione. Este sufijo incluye el ID de zona de disponibilidad de la que haya elegido.
>
> Al crear un bucket de directorio mediante una API, en la solicitud debe proporcionar el sufijo completo, incluido el ID de la zona de disponibilidad. Para obtener una lista de los ID de las zonas de disponibilidad, consulte Zonas y regiones de disponibilidad de S3 Express One Zone.

Las siguientes reglas de nomenclatura se aplican a los buckets de directorio.

- Ser únicos dentro de la Región de AWS y la zona de disponibilidad elegida.
- El nombre debe tener entre 3 (mín.) y 63 caracteres (máx.), incluido el sufijo.
- Constar de letras minúsculas, números y guiones (-).
- Comenzar y terminar por un número o una letra.
- Debe incluir el siguiente sufijo: `--azid--x-s3`.

- Los nombres de los buckets no deben comenzar con el prefijo `xn--`.
- Los nombres de los buckets no deben comenzar con el prefijo `sthree-`.
- Los nombres de los buckets no deben comenzar con el prefijo `sthree-configurator`.
- Los nombres de los buckets no deben comenzar con el prefijo `amzn-s3-demo-`.
- Los nombres de los buckets no deben terminar con el sufijo `-s3alias`. Este sufijo está reservado para nombres de alias de punto de acceso. Para obtener más información, consulte Uso de un alias de estilo de bucket para su punto de acceso de bucket de S3.
- Los nombres de los buckets no deben terminar con el sufijo `--ol-s3`. Este sufijo está reservado para nombres de alias de punto de acceso de Object Lambda. Para obtener más información, consulte Cómo usar un alias de estilo de bucket para su punto de acceso de Object Lambda de bucket de S3.
- Los nombres de los buckets no deben terminar con el sufijo `.mrap`. Este sufijo está reservado para nombres de punto de acceso de varias regiones. Para obtener más información, consulte Reglas para asignar nombres a los puntos de acceso de varias regiones de Amazon S3.

Crear un bucket de directorio

Para empezar a utilizar la clase de almacenamiento Amazon S3 Express One Zone, debe crear un bucket de directorio. La clase de almacenamiento S3 Express One Zone solo se puede usar con buckets de directorio. La clase de almacenamiento S3 Express One Zone admite casos de uso de baja latencia y proporciona un procesamiento de datos más rápido dentro de una única zona de disponibilidad. Si su aplicación es sensible al rendimiento y se beneficia de latencias `PUT` y `GET` de milisegundos de un solo dígito, le recomendamos que cree un bucket de directorio para poder utilizar la clase de almacenamiento S3 Express One Zone.

Existen dos tipos de buckets de Amazon S3: buckets de uso general y buckets de directorio. Debe elegir el tipo de bucket que mejor se adapte a sus requisitos de rendimiento y aplicación. Los buckets de uso general son del tipo de bucket de S3 original. Se recomienda usar los buckets de uso general para la mayoría de los casos de uso y patrones de acceso, pues permiten almacenar objetos en todas las clases de almacenamiento, excepto en S3 Express One Zone. Para obtener más información acerca de los buckets de uso general, consulte Descripción general de los buckets.

Los buckets de directorio utilizan la clase de almacenamiento S3 Express One Zone, que está diseñada para utilizarse en cargas de trabajo o aplicaciones fundamentales para el rendimiento que requieren una latencia uniforme en milisegundos de un solo dígito. S3 Express One Zone es la primera clase de almacenamiento de S3 en la que se puede seleccionar una única zona de

disponibilidad con la opción de coubicar su almacenamiento de objetos junto con sus recursos informáticos, lo que brinda la mayor velocidad de acceso posible. Al crear un bucket de directorio, si lo desea, puede especificar una Región de AWS y una zona de disponibilidad local para sus instancias de computación de Amazon EC2, Amazon Elastic Kubernetes Service o Amazon Elastic Container Service (Amazon ECS) a fin de optimizar el rendimiento.

Con S3 Express One Zone, sus datos se almacenan de forma redundante en varios dispositivos dentro de una única zona de disponibilidad. S3 Express One Zone está diseñado para ofrecer una disponibilidad del 99,95 % dentro de una única zona de disponibilidad y está respaldado por el contrato de nivel de servicio de Amazon S3. Para obtener más información, consulte Una sola zona de disponibilidad

Los buckets de directorio organizan los datos jerárquicamente en directorios, a diferencia de la estructura de almacenamiento plana de los buckets de uso general. No hay límites de prefijos para los buckets de directorio y los directorios individuales pueden realizar un escalado horizontal.

Para obtener más información acerca de estos buckets de directorio, consulte Buckets de directorio.

Nombres de los buckets de directorio

Los nombres de los buckets de directorio deben seguir este formato y cumplir con las reglas de denominación de los buckets de directorio:

```
bucket-base-name--azid--x-s3
```

Por ejemplo, el siguiente nombre del bucket de directorio contiene el ID de zona de disponibilidad `usw2-az1`:

```
bucket-base-name--usw2-az1--x-s3
```

Para obtener más información acerca de las reglas de nomenclatura de los bucket de directorio, consulte Reglas de nomenclatura de buckets de directorio.

Uso de la consola de S3

1. Inicie sesión en la AWS Management Console y abra la consola de Amazon S3 en https://console.aws.amazon.com/s3/.
2. En la barra de navegación de la parte superior de la página, elija el nombre de la Región de AWS que aparece. A continuación, elija en Region (Región) la región en la que desea crear un bucket.

> **Note**
>
> Puede seleccionar una región cercana para minimizar la latencia y los costos, así como para satisfacer los requisitos normativos. Los objetos almacenados en una región nunca abandonarán esa región salvo que usted los transfiera de forma específica a otra. Para una lista de Regiones de AWS de Amazon S3, consulte Puntos de conexión de Servicio de AWS en la Referencia general de Amazon Web Services.

3. En el panel de navegación izquierdo, elija Instancias.
4. Elija Crear bucket.

   Se abrirá la página Crear bucket.
5. En Configuración general, vea la Región de AWS donde se creará el bucket.
6. En Bucket type (Tipo de bucket), elija Directory (Directorio).

> **Note**
>
> - Si ha elegido una región que no admite buckets de directorio, la opción Tipo de bucket desaparece y el tipo de bucket pasa a ser un bucket de uso general de forma predeterminada. Para crear un bucket de directorio, debe elegir una región compatible. Para obtener una lista de las regiones que admiten buckets de directorio y la clase de almacenamiento Amazon S3 Express One Zone, consulte the section called “Zonas y regiones de disponibilidad de S3 Express One Zone”.
> - No se puede cambiar el tipo de bucket después de haberlo creado.

En Zona de disponibilidad, elija una zona de disponibilidad local para sus servicios de computación. Para obtener una lista de las zonas de disponibilidad que admiten buckets de directorio y la clase de almacenamiento S3 Express One Zone, consulte the section called “Zonas y regiones de disponibilidad de S3 Express One Zone”.

> **Note**
>
> La zona de disponibilidad no se puede cambiar una vez creado el bucket.

7. En Zona de disponibilidad, selecciona la casilla para confirmar que, en caso de que se produzca una interrupción en la zona de disponibilidad, es posible que sus datos no estén disponibles o se pierdan.

> **Important**
>
> Los buckets de directorio se almacenan en varios dispositivos dentro de una única zona de disponibilidad, pero no almacenan datos de forma redundante en todas las zonas de disponibilidad.

8. En Nombre del bucket, escriba un nombre para el bucket de directorio.

   Las siguientes reglas de nomenclatura se aplican a los buckets de directorio.

   - Ser únicos dentro de la Región de AWS y la zona de disponibilidad elegida.
   - El nombre debe tener entre 3 (mín.) y 63 caracteres (máx.), incluido el sufijo.
   - Constar de letras minúsculas, números y guiones (-).
   - Comenzar y terminar por un número o una letra.
   - Debe incluir el siguiente sufijo: `--`*`azid`*`--x-s3`.
   - Los nombres de los buckets no deben comenzar con el prefijo `xn--`.
   - Los nombres de los buckets no deben comenzar con el prefijo `sthree-`.
   - Los nombres de los buckets no deben comenzar con el prefijo `sthree-configurator`.
   - Los nombres de los buckets no deben comenzar con el prefijo `amzn-s3-demo-`.
   - Los nombres de los buckets no deben terminar con el sufijo `-s3alias`. Este sufijo está reservado para nombres de alias de punto de acceso. Para obtener más información, consulte Uso de un alias de estilo de bucket para su punto de acceso de bucket de S3.
   - Los nombres de los buckets no deben terminar con el sufijo `--ol-s3`. Este sufijo está reservado para nombres de alias de punto de acceso de Object Lambda. Para obtener más información, consulte Cómo usar un alias de estilo de bucket para su punto de acceso de Object Lambda de bucket de S3.
   - Los nombres de los buckets no deben terminar con el sufijo `.mrap`. Este sufijo está reservado para nombres de punto de acceso de varias regiones. Para obtener más información, consulte Reglas para asignar nombres a los puntos de acceso de varias regiones de Amazon S3.

Se añade automáticamente un sufijo al nombre base que proporciona cuando crea un bucket de directorio utilizando la consola. Este sufijo incluye el ID de zona de disponibilidad de la que haya elegido.

Una vez que haya creado el bucket, no podrá modificar su nombre. Para obtener más información sobre la nomenclatura de los buckets, consulte Reglas de nomenclatura de buckets.

> **Important**
>
> No incluya información confidencial, como números de cuenta, en el nombre del bucket. El nombre del bucket será visible en las URL que señalan a los objetos almacenados en él.

9. En Propiedad del objeto, la opción Aplicada al propietario del bucket se activa de forma automática y se desactivan todas las listas de control de acceso (ACL). En el caso de los buckets de directorio, las ACL no se pueden habilitar.

   ACL desactivadas

   - Propietario del bucket obligatorio (predeterminado): las ACL están desactivadas y el propietario del bucket tiene automáticamente la propiedad y el control total sobre cada objeto del bucket. Las ACL ya no afectan a los permisos de acceso de los datos del bucket de S3. El bucket utiliza políticas exclusivamente para definir el control de acceso.

     La mayoría de los casos de uso modernos de Amazon S3 ya no requieren el uso de ACL. Para obtener más información, consulte Control de la propiedad de los objetos y desactivación de las ACL del bucket.

10. En Configuración de bloqueo de acceso público para este bucket, tenga en cuenta que la configuración Bloquear acceso público para su bucket de directorio se habilita automáticamente. Esta configuración no se puede modificar para los bucket de directorio. Para obtener más información acerca del bloqueo del acceso público, consulte Bloquear el acceso público a su almacenamiento de Amazon S3.
11. En Configuración del cifrado del lado del servidor, Amazon S3 aplica el cifrado del lado del servidor con claves administradas por Amazon S3 (SSE-S3) como nivel básico de cifrado para todos los buckets de S3. Todas las cargas de objetos en buckets de directorio se cifran con SSE-S3. No se puede modificar el tipo de cifrado para los bucket de directorio. Para obtener

más información sobre SSE-S3, consulte the section called “Claves de cifrado administradas por Amazon S3 (SSE-S3)”.

12. Elija Crear bucket.

    Después de crear el bucket, puede añadir archivos y carpetas al bucket. Para obtener más información, consulte the section called “Trabajar con objetos en un bucket de directorio”.

Uso de los AWS SDK

SDK for Go

En este ejemplo se muestra cómo crear un bucket de directorio con el 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

En este ejemplo se muestra cómo crear un bucket de directorio con el 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

En este ejemplo se muestra cómo crear un bucket de directorio con el 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

En este ejemplo se muestra cómo crear un bucket de directorio con el 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

En este ejemplo se muestra cómo crear un bucket de directorio con el 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

En este ejemplo se muestra cómo crear un bucket de directorio con el 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

En este ejemplo se muestra cómo crear un bucket de directorio con el 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' }
  }
)
```

Uso de la AWS CLI

En este ejemplo se muestra cómo crear un bucket de directorio con el AWS CLI. Para usar el comando, sustituya los *marcadores de posición de entrada del usuario* con su propia información.

Al crear un bucket de directorio, debe proporcionar los detalles de configuración y utilizar la siguiente convención de nomenclatura: *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
```

Para obtener más información, consulte create-bucket en AWS Command Line Interface.

Visualización de las propiedades del bucket de directorio

Puede ver y configurar las propiedades de un bucket de directorio de Amazon S3 mediante la consola de Amazon S3. Para obtener más información, consulte Buckets de directorio y ¿Qué es S3 Express One Zone?.

Uso de la consola de S3

1. Inicie sesión en la AWS Management Console y abra la consola de Amazon S3 en https://console.aws.amazon.com/s3/.
2. En el panel de navegación izquierdo, elija Instancias.
3. Seleccione la pestaña Buckets de directorio.
4. En la lista Buckets de directorio, seleccione el nombre del bucket cuyas propiedades desea ver.
5. Elija la pestaña Propiedades.
6. En la pestaña Propiedades, puede ver las siguientes propiedades para el bucket:

    - Descripción general del bucket de directorio: puede ver la Región de AWS, la zona de disponibilidad, el Nombre de recurso de Amazon (ARN) y la fecha de creación del bucket.

- Cifrado predeterminado: Amazon S3 aplica el cifrado del servidor con claves administradas por Amazon S3 (SSE-S3) como el nivel básico de cifrado para cada bucket de Amazon S3. Esta opción no se puede modificar para los bucket de directorio. Amazon S3 cifra un objeto antes de guardarlo en un disco y descifra el objeto al descargarlo. Para obtener más información, consulte Establecer el comportamiento del cifrado predeterminado del lado del servidor para los buckets de Amazon S3.

  Para obtener más información acerca de las características admitidas para los bucket de directorio, consulte Características de S3 Express One Zone.

Administración de políticas de buckets para buckets de directorio

Puede añadir, eliminar, actualizar y ver políticas de buckets de directorio de Amazon S3 desde la consola de Amazon S3 y los SDK de AWS. Para obtener más información, consulte los siguientes temas. Para obtener más información sobre las acciones de AWS Identity and Access Management (IAM) y las claves de condición admitidas para S3 Express One Zone, consulte AWS Identity and Access Management (IAM) para S3 Express One Zone. Para ver ejemplos de políticas de bucket para buckets de directorio, consulte Ejemplos de políticas de bucket de directorio para S3 Express One Zone.

Temas

- Agregar una política de bucket
- Visualización de una política de bucket
- Eliminación de una política de bucket

Agregar una política de bucket

Para agregar una política de bucket a un bucket de directorio, puede utilizar la consola de Amazon S3 o los SDK de AWS o la AWS CLI.

Uso de la consola de S3

Para crear o editar una política de bucket

1. Inicie sesión AWS Management Console Management Console y abra la consola de Amazon S3 en https://console.aws.amazon.com/s3/.
2. En el panel de navegación izquierdo, elija Instancias.

3. Seleccione la pestaña Buckets de directorio.
4. En la lista Buckets, elija el nombre del bucket en el que desea cargar sus carpetas o archivos.
5. Elija la pestaña Permisos.
6. En Política de bucket, elija Editar. Aparece la página Editar política de bucket.
7. Para generar una política automáticamente, elija Generador de políticas.

   Si elige Generador de políticas, se abre el generador de políticas de AWS en una ventana nueva.

   Si no quiere usar el AWSgenerador de políticas, puede añadir o editar las instrucciones JSON en la sección Política.

   a. En la página Generador de políticas de AWS, para Seleccionar tipo de política, elija Política de bucket de S3.
   b. Agregue una instrucción ingresando la información en los campos proporcionados y, a continuación, elija Agregar declaración. Repita este paso para todas las instrucciones que desee agregar. Para obtener más información acerca de estos campos, consulte la Referencia de los elementos de las políticas de JSON de IAM en la Guía del usuario de IAM.

      > **Note**
      >
      > Para mayor comodidad, la página Editar la política del bucket muestra el ARN del bucket (Nombre de recurso de Amazon) actual encima del campo de texto Política. Puede copiar este ARN para utilizarlo en las instrucciones de la página Generador de políticas de AWS.

   c. Una vez que haya terminado de agregar instrucciones, elija Generar política.
   d. Copie el texto de la política generada, elija Cerrar y vuelva a la página Editar política de bucket en la consola de Amazon S3.
8. En el cuadro Política, edite la política existente o pegue la política de bucket desde el generador de políticas de AWS. Asegúrese de resolver advertencias de seguridad, errores, advertencias generales y sugerencias antes de guardar la política.

> **Note**
>
> Las políticas de bucket tienen un límite de tamaño de 20 KB.

9. Elija Guardar cambios, que le devuelve a la pestaña Permisos.

Uso de los 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);
        }
    }
```

Uso de la AWS CLI

En este ejemplo se muestra cómo agregar un bucket de política a un bucket de directorio con la AWS CLI. Para usar el comando, sustituya los *marcadores de posición de entrada del usuario* con su propia información.

```
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/"
        }
    ]
}
```

Para obtener más información, consulte put-bucket-policy en AWS Command Line Interface.

Visualización de una política de bucket

Para ver una política de buckets para un bucket de directorio, utilice los siguientes ejemplos.

Uso de la AWS CLI

En este ejemplo se muestra cómo ver una politica de bucket adjunta a un bucket de directorio con la AWS CLI. Para usar el comando, sustituya los *marcadores de posición de entrada del usuario* con su propia información.

```
aws s3api get-bucket-policy --bucket bucket-base-name--azid--x-s3
```

Para obtener más información, consulte get-bucket-policy en AWS Command Line Interface.

Eliminación de una política de bucket

Para eliminar una política de bucket para un directorio de bucket, utilice los siguientes ejemplos.

Uso de los 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);
      }
```

Uso de la AWS CLI

En este ejemplo se muestra cómo eliminar una política de bucket para un bucket de directorio con la AWS CLI. Para usar el comando, sustituya los *marcadores de posición de entrada del usuario* con su propia información.

```
aws s3api delete-bucket-policy --bucket bucket-base-name--azid--x-s3
```

Para obtener más información, consulte delete-bucket-policy en AWS Command Line Interface.

Vaciado de un bucket de directorio

Puede vaciar un bucket de directorio de Amazon S3 con la consola de Amazon S3. Para obtener más información acerca de estos buckets de directorio, consulte Buckets de directorio.

Antes de vaciar un bucket de directorio, tenga en cuenta lo siguiente:

- Cuando vacía un bucket de directorio, elimina todos los objetos, pero conserva el bucket de directorio.
- Después de vaciar un bucket de directorio, la acción de vaciado no se puede deshacer.
- Es posible que se eliminen los objetos agregados al bucket de directorio mientras la acción de vaciado del bucket esté en curso.

Si también quiere eliminar el bucket, tenga en cuenta lo siguiente:

- Todos los objetos del bucket de directorio deben eliminarse antes de eliminar el propio bucket:
- Las cargas multiparte en curso en el bucket de directorio se deben cancelar antes de poder eliminar el propio bucket.

> **Note**
>
> El comando `s3 rm` mediante la AWS Command Line Interface (CLI), la operación `delete` mediante Mountpoint y el botón de opciones del bucket Vacío mediante la AWS Management Console no pueden eliminar las cargas multiparte en curso en un bucket de directorio. Para eliminar estas cargas multiparte en curso, utilice la operación `ListMultipartUploads` para mostrar las cargas multiparte en curso en el bucket y utilice la operación `AbortMultipartUpload` para anular todas las cargas multiparte en curso.

Para eliminar un bucket de directorio, consulte Eliminar un bucket de directorio. Para anular una carga multiparte en curso, consulte the section called “Anulación de la carga multiparte”.

Para vaciar un bucket de uso general, consulte Vaciar un bucket.

Uso de la consola de S3

Cómo vaciar un bucket de directorio

1. Inicie sesión AWS Management Console Management Console y abra la consola de Amazon S3 en https://console.aws.amazon.com/s3/.
2. En el panel de navegación izquierdo, elija Instancias.
3. Seleccione la pestaña Buckets de directorio.
4. Elija el botón de opción junto al nombre del bucket que desea vaciar y, a continuación, Vaciar.
5. En la página Empty bucket (Vaciar bucket), confirme que desea vaciar el bucket; para ello, ingrese **`permanently delete`** en el campo de texto y luego, elija Empty (Vaciar).
6. Monitoree el progreso del proceso de vaciado del bucket en la página Vaciar bucket: estado.

Eliminar un bucket de directorio

Solo se pueden eliminar buckets de directorio de Amazon S3. Antes de eliminar el bucket de directorio, debe eliminar todos los objetos del bucket y anular todas las cargas multiparte en curso.

Para vaciar un bucket de directorio, consulte Vaciado de un bucket de directorio. Para anular una carga multiparte en curso, consulte the section called “Anulación de la carga multiparte”.

Para eliminar un bucket de uso general, consulte Eliminar un bucket.

Uso de la consola de S3

Después de vaciar el bucket de directorio y anular todas las cargas multiparte en curso, ya puede eliminar el bucket.

1. Inicie sesión AWS Management Console Management Console y abra la consola de Amazon S3 en https://console.aws.amazon.com/s3/.
2. En el panel de navegación izquierdo, elija Instancias.
3. Seleccione la pestaña Buckets de directorio.

4. En la lista Buckets de directorio, elija el botón de opción junto al bucket que desea eliminar.
5. Elija Eliminar.
6. En la página Eliminar bucket, escriba el nombre del bucket en el campo de texto para confirmar la eliminación del bucket.

> **Important**
>
> La eliminación de un bucket de directorio no se puede revertir.

7. Para eliminar el bucket de directorio, elija Eliminar bucket.

Uso de los SDK de AWS

En los siguientes ejemplos se elimina un bucket de directorio con el AWS SDK for Java 2.x y el 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)
```

Uso de la AWS CLI

En este ejemplo se muestra cómo crear un bucket de directorio con la AWS CLI. Para usar el comando, sustituya los *marcadores de posición de entrada del usuario* con su propia información.

```
aws s3api delete-bucket --bucket bucket-base-name--azid--x-s3 --region us-west-2
```

Para obtener más información, consulte Eliminación de un bucket en AWS Command Line Interface.

Mostrar una lista de buckets de directorio

En los siguientes ejemplos se muestra cómo obtener una lista de buckets de directorio con los SDK de AWS y la CLI de AWS.

Uso de los SDK de AWS

SDK for Java 2.x

Example

En el siguiente ejemplo se muestra una lista de buckets de directorio mediante el 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

En el siguiente ejemplo se muestra una lista de buckets de directorio mediante el 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

En el siguiente ejemplo se muestra una lista de buckets de directorio mediante el AWS SDK for .NET.

```
var listDirectoryBuckets = await amazonS3Client.ListDirectoryBucketsAsync(new
 ListDirectoryBucketsRequest
{
  MaxDirectoryBuckets = 10
  }).ConfigureAwait(false);
```

SDK for PHP

Example

En el siguiente ejemplo se muestra una lista de buckets de directorio mediante el AWS SDK for PHP.

```
require 'vendor/autoload.php';
```

```
$s3Client = new S3Client([
    'region'      => 'us-east-1',
]);
$result = $s3Client->listDirectoryBuckets();
```

SDK for Ruby

Example

En el siguiente ejemplo se muestra una lista de buckets de directorio mediante el AWS SDK for Ruby.

```
s3 = Aws::S3::Client.new(region:'us-west-1')
s3.list_directory_buckets
```

Uso de la AWS CLI

El siguiente comando de ejemplo `list-directory-buckets` muestra cómo puede usar la AWS CLI para obtener una lista de tus buckets de directorio en la región *us-east-1*. Para ejecutar este comando, sustituya los *user input placeholders* con su propia información.

```
aws s3api list-directory-buckets --region us-east-1
```

Para obtener más información, consulte list-directory-buckets en la Referencia de los comandos de AWS CLI.

Uso de **HeadBucket** con buckets de directorio

En los siguientes ejemplos del SDK de AWS se muestra cómo usar la operación de la API `HeadBucket` para determinar si existe un bucket de Amazon S3 y si tiene permiso para acceder a él.

Uso de los SDK de AWS

En el siguiente ejemplo de AWS SDK for Java 2.x se muestra cómo determinar si existe un bucket y si tiene permiso para acceder a él.

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);
   }
}
```

Uso de la AWS CLI

En el siguiente ejemplo de `head-bucket` se muestra cómo usar la AWS CLI para determinar si existe un bucket de directorio y si tiene permiso para acceder a él. Para ejecutar este comando, sustituya los marcadores de posición de entrada del usuario con su propia información.

```
aws s3api head-bucket --bucket bucket-base-name--azid--x-s3
```

Para obtener más información, consulte head-bucket en la Referencia de los comandos de AWS CLI.

Uso de objetos en un bucket de directorio

Tras crear un bucket de directorio de Amazon S3, puede trabajar con objetos desde la consola de Amazon S3, la AWS Command Line Interface (AWS CLI) y los SDK de AWS.

Para obtener más información sobre las operaciones en masa con objetos almacenados en la clase de almacenamiento S3 Express One Zone, consulte Administración de objetos. Para obtener más información sobre cómo importar, cargar, copiar, eliminar y descargar objetos y leer metadatos de objetos de buckets de directorio, consulte los siguientes temas.

Temas

- Importación de objetos a un bucket de directorio

- Uso de operaciones por lotes con S3 Express One Zone
- Cargar un objeto en un bucket de directorio
- Uso de las cargas multiparte con buckets de directorio
- Copiar un objeto en un bucket de directorio
- Eliminación de un objeto de un bucket de directorio
- Descarga de un objeto en un bucket de directorio
- Uso de HeadObject con buckets de directorio

Importación de objetos a un bucket de directorio

Tras crear un bucket de directorio en Amazon S3, puede rellenar el nuevo bucket con datos mediante la acción de importación. La importación es un método simplificado para crear trabajos de Operaciones por lotes de S3 a fin de copiar objetos de buckets de uso general a buckets de directorio.

> **Note**
>
> Se aplican a la importación de trabajos las limitaciones siguientes:
>
> - El bucket de origen y el bucket de destino deben estar en la misma cuenta y Región de AWS.
> - El bucket de origen no puede ser un bucket de directorio.
> - Los objetos de más de 5 GB no son compatibles y se omitirán en la operación de copia.
> - Los objetos de las clases de almacenamiento de los niveles Glacier Flexible Retrieval, Glacier Deep Archive, Intelligent-Tiering Archive Access e Intelligent-Tiering Deep Archive deben restaurarse antes de poder importarlos.
> - Los objetos importados con algoritmos de suma de comprobación MD5 se convierten para usar sumas de comprobación CRC32.
> - Los objetos importados utilizan cifrado del servidor con claves administradas de Amazon S3 (SSE-S3)
> - Los objetos importados utilizan la clase de almacenamiento Express One Zone, que tiene una estructura de precios diferente a la de las clases de almacenamiento que se utilizan en los buckets de uso general. Tenga en cuenta esta diferencia de coste al importar un gran número de objetos.

Al configurar un trabajo de importación, se especifica el bucket o prefijo de origen desde el que se copiarán los objetos existentes. También proporciona un rol de AWS Identity and Access Management (IAM) con permisos para acceder a los objetos de origen. A continuación, Amazon S3 inicia un trabajo de Operaciones por lotes que copia los objetos y aplica automáticamente la clase de almacenamiento y la configuración de suma de comprobación correspondientes.

Utilice la consola de Amazon S3 para configurar trabajos de importación.

Uso de la consola de Amazon S3

Para importar objetos a un bucket de directorio

1. Inicie sesión AWS Management Console Management Console y abra la consola de Amazon S3 en https://console.aws.amazon.com/s3/.
2. En el panel de navegación de la izquierdo, seleccione Buckets y, a continuación, la pestaña Buckets de directorio. Seleccione el botón de opción junto al bucket de directorio al que desea importar objetos.
3. Seleccione Importar.
4. En Origen, introduzca el bucket de uso general (o la ruta del bucket, incluido el prefijo) que contiene los objetos que desea importar. Para elegir un bucket de uso general existente de una lista, seleccione Examinar S3.
5. En Permiso para acceder a los objetos de origen y copiarlos, realice una de las siguientes operaciones para especificar un rol de IAM con los permisos necesarios para importar los objetos de origen:

   - Para que Amazon S3 pueda crear un nuevo rol de IAM en su nombre, seleccione Crear un nuevo rol de IAM.

     > **Note**
     >
     > Si los objetos de origen se cifran en el servidor con claves de AWS Key Management Service (AWS KMS) (SSE-KMS), no elija la opción Crear un nuevo rol de IAM. En su lugar, especifique un rol de IAM existente que tenga el permiso `kms:Decrypt`. Amazon S3 utilizará este permiso para descifrar sus objetos. Durante el proceso de importación, Amazon S3 volverá a cifrar esos objetos mediante un cifrado del servidor con claves administradas de Amazon S3 (SSE-S3).

- Para elegir un rol de IAM existente de una lista, seleccione Elegir entre los roles de IAM existentes.
- Para especificar un rol de IAM existente con su nombre de recurso de Amazon (ARN), seleccione Introducir ARN del rol de IAM y, a continuación, introduzca el ARN en el campo correspondiente.

6. Revise la información que aparece en las secciones Destino y Configuración de objetos copiada. Si la información de la sección Destino es correcta, seleccione Importar para iniciar el trabajo de copia.

   La consola de Amazon S3 muestra el estado del nuevo trabajo en la página Operaciones por lotes. Para obtener más información sobre el trabajo, pulse el botón de opción junto al nombre del trabajo y, a continuación, en el menú Acciones, seleccione Ver detalles. Para abrir el bucket de directorio al que se importarán los objetos, seleccione Ver destino de la importación.

Uso de operaciones por lotes con S3 Express One Zone

Puede utilizar operaciones por lotes de Amazon S3 para realizar operaciones en objetos almacenados en buckets de S3. Para obtener más información sobre las operaciones por lotes de S3, consulte Realización de operaciones por lotes a gran escala en objetos de Amazon S3.

En los siguientes temas se trata la realización de operaciones por lotes con objetos almacenados en buckets de directorio de la clase de almacenamiento S3 Express One Zone.

Temas

- Uso de operaciones por lotes con buckets de directorio
- Diferencias clave

Uso de operaciones por lotes con buckets de directorio

Puede realizar la operación Copiar y las operaciones Invocar función de AWS Lambda en los objetos almacenados en los buckets de directorio. Con Copiar, puede copiar objetos entre buckets del mismo tipo (por ejemplo, de un bucket de directorio a otro). También puede copiar entre buckets de uso general y buckets de directorio. Con Invocar función de AWS Lambda, puede utilizar una función de Lambda para realizar acciones en los objetos de su bucket de directorio con el código que defina.

Copia de objetos

Puede copiar entre el mismo tipo de bucket o entre buckets de directorio y buckets de uso general. Al copiar en un bucket de directorio, debe usar el formato de Nombre de recurso de Amazon (ARN) correcto para ese tipo de bucket. El formato de ARN de un bucket de directorio es `arn:aws:s3express:region:account-id:bucket/bucket-base-name--x-s3`.

También puede rellenar el bucket de directorio con datos mediante la acción Importar de la consola de S3. Importar es un método simplificado para crear trabajos de Batch Operations a fin de copiar objetos de buckets de uso general a buckets de directorio. Para Importar trabajos de copia de buckets de uso general a buckets de directorio, S3 genera automáticamente un manifiesto. Para obtener más información, consulte Importación de objetos a un bucket de directorio y Especificar un manifiesto.

Invocación de funciones de Lambda (**`LambdaInvoke`**)

Existen requisitos especiales para utilizar las operaciones por lotes para invocar funciones de Lambda que actúan en los buckets de directorio. Por ejemplo, debe estructurar la solicitud de Lambda mediante un esquema de invocación JSON v2 y especificar InvocationSchemaVersion 2.0 al crear el trabajo. Para obtener más información, consulte Invocar a la función AWS Lambda.

Diferencias clave

A continuación se muestra una lista de las principales diferencias al usar Operaciones por lotes para realizar operaciones masivas en objetos almacenados en buckets de directorio que utilizan la clase de almacenamiento S3 Express One Zone:

- Amazon S3 cifra automáticamente todos los objetos nuevos que se cargan a un bucket de S3. La configuración de cifrado predeterminada de un bucket de S3 siempre está activada y, como mínimo, se establece en el cifrado del servidor con claves administradas de Amazon S3 (SSE-S3). Para los buckets de directorio, solo se admite SSE-S3. Al realizar una solicitud `CopyObject` que establece el cifrado del servidor con claves proporcionadas por el cliente (SSE-C) o el cifrado del servidor con claves de AWS Key Management Service (AWS KMS) (SSE-KMS) en un bucket de directorio (origen o destino), la respuesta devolverá un error HTTP `400 (Bad Request)`.
- Los objetos de los buckets de directorio no se pueden etiquetar. Solo se puede especificar un conjunto de etiquetas vacío. De forma predeterminada, Operaciones por lotes copia las etiquetas. Si copia un objeto que tiene etiquetas entre los buckets de uso general y los de directorio, recibirá una respuesta `501 (Not Implemented)`.
- S3 Express One Zone le ofrece la opción de elegir el algoritmo de suma de comprobación que se utiliza para validar los datos durante las cargas o descargas. Puede seleccionar uno

de los siguientes algoritmos de comprobación de integridad de datos Secure Hash Algoritms (SHA) o Cyclic Redundancy Check (CRC): CRC32, CRC32, SHA-1 y SHA-256. Las sumas de comprobación basadas en MD5 no son compatibles con la clase de almacenamiento S3 Express One Zone.

- De forma predeterminada, todos los buckets de Amazon S3 establecen la configuración de S3 Object Ownership en Aplicada al propietario del bucket y las listas de control de acceso (ACL) están deshabilitadas. Esta opción no se puede modificar para los bucket de directorio. Puede copiar un objeto desde buckets de uso general a buckets de directorio. Sin embargo, no puede anular la ACL predeterminada al copiar a o desde un bucket de directorio.
- Independientemente de cómo especifique el manifiesto, la propia lista debe almacenarse en un bucket de uso general. Operaciones por lotes no puede importar los manifiestos existentes desde (ni guardar los manifiestos generados en) buckets de directorio. Sin embargo, los objetos descritos en el manifiesto se pueden almacenar en buckets de directorio.
- Operaciones por lotes no puede especificar un bucket de directorio como ubicación en un informe de inventario de S3. Los informes de inventario no admiten buckets de directorio. Para crear un archivo de manifiesto para los objetos dentro de un bucket de directorio, use la operación de la API `ListObjectsV2` para enumerar los objetos. A continuación, inserte la lista en un archivo CSV.

Concesión de acceso

Para realizar trabajos de copia, debe tener los siguientes permisos:

- Para copiar objetos de un bucket de directorio a otro bucket de directorio, debe disponer del permiso `s3express:CreateSession`.
- Para copiar objetos de buckets de directorio a buckets de uso general, debe tener el permiso `s3express:CreateSession` y el permiso `s3:PutObject` para escribir la copia del objeto en el bucket de destino.
- Para copiar objetos de buckets de uso general en buckets de directorio, debe tener el permiso `s3express:CreateSession` y el permiso `s3:GetObject` para leer el objeto de origen que se está copiando.

  Para obtener más información, consulte CopyObject en la Referencia de la API de Amazon Simple Storage Service.
- Para invocar una función de Lambda, debe conceder permisos al recurso según la función de Lambda. Compruebe los permisos de la API correspondientes para saber qué permisos son necesarios.

Cargar un objeto en un bucket de directorio

Después de crear un bucket de directorio de Amazon S3, puede cargar objetos en él. En estos ejemplos se muestra cómo cargar un objeto en un bucket de directorio mediante la consola de S3 y los SDK de AWS. Para obtener más información sobre las operaciones de carga de objetos de forma masiva con S3 Express One Zone, consulte Administración de objetos.

Uso de la consola de S3

1. Inicie sesión en la AWS Management Console y abra la consola de Amazon S3 en https://console.aws.amazon.com/s3/.
2. En el panel de navegación izquierdo, elija Instancias.
3. Seleccione la pestaña Buckets de directorio.
4. Elija el nombre del bucket en el que desea cargar sus carpetas o archivos.
5. En la pestaña Objetos, elija Cargar.
6. En la página Cargar, lleve a cabo alguna de las siguientes acciones:
   - Arrastre y suelte archivos y carpetas en el área de carga punteada.
   - Elija Agregar archivos o Agregar carpeta, elija los archivos o carpetas que desee cargar y, a continuación, seleccione Abrir o Cargar.
7. En Sumas de comprobación, seleccione la función de suma de comprobación que desee usar.

   (Opcional) Si va a cargar un único objeto de un tamaño inferior a 16 MB, también puede especificar un valor de suma de comprobación precalculado. Al proporcionar un valor precalculado, Amazon S3 lo compara con el valor que calcula mediante la función de suma de comprobación seleccionada. Si los valores no coinciden, la carga no se iniciará.
8. Las opciones de las secciones Permisos y Propiedades se configuran automáticamente con la configuración predeterminada y no se pueden modificar. El bloqueo de acceso público se habilita automáticamente. El control de versiones de S3 y el bloqueo de objetos de S3 no se pueden habilitar en los buckets de directorio.

   (Opcional) Si quiere añadir metadatos en pares clave-valor a sus objetos, expanda la sección Propiedades y, a continuación, en la sección Metadatos, elija Agregar metadatos.
9. Para cargar inmediatamente los archivos y carpetas indicados, elija Cargar.

   Amazon S3 carga sus objetos y carpetas. Cuando finalice la carga, puede ver un mensaje de éxito en la página Cargar: estado.

Uso de los SDK de AWS

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()
```

Uso de la AWS CLI

En el siguiente comando de ejemplo de `put-object`, se muestra cómo puede utilizar la AWS CLI para cargar un objeto de Amazon S3. Para ejecutar este comando, sustituya los *`user input placeholders`* con su propia información.

```
aws s3api put-object --bucket bucket-base-name--azid--x-s3 --key sampleinut/file001.bin
 --body bucket-seed/file001.bin
```

Para obtener más información, consulte put-object en la Referencia de los comandos de AWS CLI.

Uso de las cargas multiparte con buckets de directorio

Puede usar el proceso de carga multiparte para cargar un solo objeto como un conjunto de partes. Cada parte es una parte contigua de los datos del objeto. Puede cargar estas partes del objeto de forma independiente y en cualquier orden. Si la transmisión de cualquier parte falla, puede retransmitir esta parte sin que las demás partes se vean afectadas. Después de cargar todas las partes del objeto, Amazon S3 las combina y crea el objeto. Por lo general, cuando el tamaño del

objeto alcanza los 100 MB, deberá usar las cargas multipartes en lugar de cargar el objeto en una única operación.

El uso de la carga multiparte proporciona las siguientes ventajas:

- Mayor rendimiento: puede cargar las partes al mismo tiempo para aumentar el rendimiento.
- Recuperación rápida ante cualquier problema de red: una parte de tamaño más pequeño minimiza el impacto de tener que reiniciar una carga fallida debido a un error de red.
- Detención y reanudación de cargas de objetos: puede cargar las partes del objeto con el paso del tiempo. Después de iniciar una carga multiparte, no hay fecha de caducidad. Debe completar o anular la carga multiparte de forma explícita.
- Inicio de una carga antes de conocer el tamaño final del objeto: puede cargar un objeto a medida que lo crea.

Le recomendamos que use las cargas multiparte de las siguientes maneras:

- Si carga objetos grandes en una red de banda ancha estable, use las cargas multiparte para aumentar al máximo el uso del ancho de banda disponible cargando las partes de objetos en paralelo para obtener un rendimiento en varios subprocesos.
- Si realiza la carga en una red irregular, use las cargas multiparte para aumentar la resiliencia ante errores de red para evitar reiniciar la carga. Al usar las cargas multiparte, debe volver a intentar cargar solo las partes que se han interrumpido durante la carga. No necesita reiniciar la carga de su objeto desde el principio.

Si utiliza cargas multiparte para cargar objetos a la clase de almacenamiento Amazon S3 Express One Zone en buckets de directorio, el proceso de carga multiparte es similar al proceso de utilizar la carga multiparte para cargar objetos en buckets de uso general. Sin embargo, hay algunas diferencias notables.

Para obtener más información sobre el uso de las cargas multiparte para cargar objetos en S3 Express One Zone, consulte los siguientes temas.

Temas

- Cargas multiparte y precios
- Operaciones de la API y permisos de carga multiparte
- Ejemplos

El proceso de carga multiparte

Una carga multiparte es un proceso de tres pasos:

- Usted inicia la carga.
- Usted carga las partes del objeto.
- Después de cargar todas las partes, completa la carga multiparte.

Al recibir la solicitud de carga multiparte completa, Amazon S3 construye el objeto a partir de las partes cargadas para que pueda obtener acceso al objeto como lo haría con cualquier otro objeto de su bucket.

Inicio de la carga multiparte

Al enviar una solicitud para iniciar una carga multiparte, Amazon S3 devuelve una respuesta con un ID de carga, que es un identificador único para su carga multiparte. Debe incluir este ID de carga siempre que cargue las partes, enumera las partes, completa una carga o la anula.

Carga de partes

Al cargar una parte, además del ID de carga, debe especificar un número de parte. Al utilizar una carga multiparte con S3 Express One Zone, los números de las partes multiparte deben ser consecutivos. Si intenta completar una solicitud de carga multiparte con números de partes no consecutivos, se genera un error `HTTP 400 Bad Request` (número de parte no válido).

Un número de parte identifica una parte de forma exclusiva y su posición en el objeto que se está cargando. Si carga una parte nueva con el mismo número que una parte ya cargada, se anulará la parte existente.

Siempre que cargue una parte, Amazon S3 devolverá un encabezado de etiqueta de entidad (ETag) en la respuesta. Para cada carga de parte, debe anotar el número de parte y el valor de ETag. Los valores de ETag para todas las cargas de partes de objeto seguirán siendo los mismos, pero a cada

parte se le asignará un número de pieza diferente. Debe incluir estos valores en la solicitud posterior para completar la carga multiparte.

Amazon S3 cifra automáticamente todos los objetos nuevos que se cargan a un bucket de S3. Al realizar una carga multiparte, si no especifica la información de cifrado en su solicitud, la configuración de cifrado de las partes cargadas se establece con la configuración de cifrado predeterminada del bucket de destino. La configuración de cifrado predeterminada de un bucket de Amazon S3 siempre está habilitada y, como mínimo, se establece para el cifrado del servidor con claves administradas de Amazon S3 (SSE-S3). Para los buckets de directorio, solo se admite SSE-S3. Para obtener más información, consulte Cifrado del servidor con claves administradas por Amazon S3 (SSE-S3).

Finalización de la carga multiparte

Al completar una carga multiparte, Amazon S3 crea un objeto al concatenar las partes en orden ascendente según el número de parte. Después de una solicitud de completar realizada correctamente, las partes ya no existirán.

La solicitud carga multiparte completa debe incluir el ID de carga y una lista de ambos números de parte y sus valores ETag correspondientes. La respuesta de Amazon S3 incluye una ETag que identifica de forma exclusiva los datos de objetos combinados. Esta ETag no es un hash de MD5 de los datos del objeto.

Listas de cargas multiparte

Puede enumerar las partes de una carga multiparte específica o todas las cargas multipartes en curso. La operación de lista de partes devuelve la información de las partes que ha cargado para una carga multiparte específica. Para cada solicitud de lista de partes, Amazon S3 devuelve la información de las partes para la carga multiparte específica, hasta un máximo de 1000 partes. Si hay más de 1000 partes en la carga multiparte, debe usar la paginación para recuperar todas las partes.

La lista de partes que se devuelve no incluye las partes que no se hayan cargado por completo. Con la operación lista de cargas multiparte, puede obtener una lista de las cargas multiparte en curso.

Una carga multiparte en curso es una carga iniciada pero que aún no se ha completado ni anulado. Cada solicitud devuelve 1 000 cargas multipartes como máximo. Si hay más de 1 000 cargas multiparte en curso, debe enviar otras solicitudes para recuperar las cargas multiparte restantes. Solo utilice la lista devuelta para fines de verificación. No utilice el resultado de esta lista al enviar

una solicitud para completar la carga multiparte. En cambio, mantenga su propia lista de números de parte que especificó al cargarlas y los valores correspondientes de ETag que devuelve Amazon S3.

Para obtener más información sobre las listas de carga multiparte, consulte ListParts en la Referencia de la API de Amazon Simple Storage Service.

Sumas de comprobación con operaciones de carga multiparte

Al cargar un objeto, es posible especificar un algoritmo de suma de comprobación para comprobar la integridad del objeto. Los buckets de directorio no admiten MD5. Puede especificar uno de los siguientes algoritmos de comprobación de integridad de datos Secure Hash Algoritms (SHA) o Cyclic Redundancy Check (CRC):

- CRC32
- CRC 32C
- SHA-1
- SHA-256

Puede utilizar la API de REST de Amazon S3 o el SDK de AWS para obtener el valor de la suma de comprobación de partes individuales mediante `GetObject` o `HeadObject`. Si quiere recuperar los valores de la suma de comprobación de partes individuales de las cargas multiparte que aún están en proceso, puede utilizar `ListParts`.

> **Important**
>
> Al utilizar los algoritmos de suma de comprobación anteriores, los números de partes multiparte deben ser consecutivos. Si intenta completar una solicitud de carga multiparte con números de partes no consecutivos, Amazon S3 genera un error `HTTP 400 Bad Request` (número de parte no válido).

Para obtener más información sobre cómo funcionan las sumas de comprobación con objetos multiparte, consulte Comprobación de la integridad de objetos.

Operaciones de carga multiparte simultáneas

En un entorno de desarrollo distribuido, es posible que la aplicación inicie varias actualizaciones en el mismo objeto simultáneamente. Por ejemplo, la aplicación puede iniciar varias cargas multiparte

con la misma clave de objeto. Para cada una de estas cargas, la aplicación puede cargar las partes y enviar una solicitud de carga completa a Amazon S3 para crear el objeto. Para S3 Express One Zone, la hora de creación del objeto es la fecha de finalización de la carga multiparte.

> **Important**
>
> No se admite el control de versiones para los objetos almacenados en los buckets de directorio.

Cargas multiparte y precios

Después de iniciar una carga multiparte, Amazon S3 retiene todas las partes hasta que complete o anule la carga. Durante la vida útil, se le cobrará por todo el almacenamiento, el ancho de banda y las solicitudes para esta carga multiparte y sus partes asociadas. Si anula la carga multiparte, no se le cobrará por estos elementos, ya que Amazon S3 elimina los artefactos cargados y cualquier parte que haya cargado. No se cobran gastos de eliminación anticipada por eliminar cargas multiparte incompletas, independientemente de la clase de almacenamiento especificada. Para obtener más información acerca de los precios, consulte Precios de Amazon S3.

> **Important**
>
> Si la solicitud de carga multiparte completa no se envía correctamente, las partes del objeto no se ensamblan y no se crea ningún objeto. Se facturará todo el almacenamiento asociado con las partes cargadas. Es importante que complete la carga multiparte para que se cree el objeto o anule la carga multiparte a fin de eliminar las partes cargadas.
> Antes de eliminar un bucket de directorio, debe completar o anular todas las cargas multiparte en curso. Los buckets de directorio no admiten las configuraciones de S3 Lifecycle. Si es necesario, puede elaborar una lista de cargas multiparte activas, anular las cargas y, por último, eliminar el bucket.

Operaciones de la API y permisos de carga multiparte

Para permitir el acceso a las operaciones de la API de administración de objetos en un bucket de directorio, debe conceder el permiso `s3express:CreateSession` en una política de bucket o una política de AWS Identity and Access Management (IAM) basada en identidades.

Debe tener los permisos necesarios para utilizar las operaciones de carga multiparte. Puede usar políticas de bucket o políticas de IAM basadas en identidades para conceder permisos a las entidades principales de IAM para realizar estas operaciones. En la siguiente tabla se muestran los permisos requeridos para varias operaciones de carga multiparte.

Puede identificar al iniciador de una carga multiparte mediante el elemento `Initiator`. Si el iniciador es una Cuenta de AWS, este elemento proporcionará la misma información que el elemento `0wner`. Si el iniciador es un usuario de IAM, este elemento proporciona el ARN de usuario y el nombre para mostrar.

Acción	Permisos necesarios
Crear una carga multiparte	Para poder crear la carga multiparte, debe tener permiso para realizar la acción `s3express:CreateSession` en el bucket de directorio.
Iniciar una carga multiparte	Para iniciar la carga multiparte, debe tener permiso para realizar la acción `s3express:CreateSession` en el bucket de directorio.
Cargar una parte	Para cargar una parte, debe tener permiso para realizar la acción `s3express:CreateSession` en el bucket de directorio. El propietario del bucket debe permitir al iniciador realizar la acción `s3express:CreateSession` en un bucket de directorio para que el iniciador pueda cargar la parte.
Cargar una parte (copia)	Para cargar una parte, debe tener permiso para realizar la acción `s3express:CreateSession` en el bucket de directorio. Para que el iniciador cargue una parte del objeto, el propietario del bucket debe permitir al iniciador realizar la acción `s3express:CreateSession` en un objeto.
Completar una carga multiparte	Para completar una carga multiparte, debe tener permiso para realizar la acción `s3express:CreateSession` en el bucket de directorio.

Acción	Permisos necesarios
	El propietario del bucket debe permitir al iniciador realizar la acción `s3express:CreateSession` en un objeto para que el iniciador pueda completar la carga multiparte.
Anular una carga multiparte	Para anular una carga multiparte, debe tener permiso para realizar la acción `s3express:CreateSession` . Para que pueda anular la carga multiparte, se debe conceder al iniciador un permiso de acceso explícito para realizar la acción `s3express:CreateSession` .
Enumerar las partes	Para enumerar las partes de una carga multiparte, debe tener permiso para realizar la acción `s3express:CreateSession` en el bucket de directorio.
Obtenga una lista de las cargas multiparte en curso	Para enumerar las cargas multiparte en curso de un bucket, debe tener permiso para realizar la acción `s3:ListBucketMultipartUploads` en dicho bucket.

Compatibilidad de la operación de la API con las cargas multiparte

En las siguientes secciones de la referencia de la API de Amazon Simple Storage Service se describen las operaciones de la API de REST de Amazon S3 para las cargas multiparte.

- CreateMultipartUpload
- UploadPart
- UploadPartCopy
- CompleteMultipartUpload
- AbortMultipartUpload
- ListParts
- ListMultipartUploads

Ejemplos

Para usar una carga multiparte para cargar un objeto en S3 Express One Zone en un bucket de directorio, consulte los siguientes ejemplos.

Temas

Creación de una carga multiparte

En el siguiente ejemplo, se muestra cómo detener una carga multiparte.

Uso de los SDK de AWS

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
```

Uso de la AWS CLI

En este ejemplo se muestra cómo crear una carga multiparte en un bucket de directorio con la AWS CLI. Este comando inicia una carga multiparte en el bucket de directorio *bucket-base-*

name--*azid*--x-s3 para el objeto *KEY_NAME*. Para usar el comando, sustituya los *marcadores de posición de entrada del usuario* con su propia información.

```
aws s3api create-multipart-upload --bucket bucket-base-name--azid--x-s3 --key KEY_NAME
```

Para obtener más información, consulte create-multipart-upload en la AWS Command Line Interface.

Carga de las partes de una carga multiparte

En los siguientes ejemplos de código se muestra cómo cargar partes de una carga multiparte.

Uso de los SDK de AWS

SDK for Java 2.x

En los siguientes ejemplos se muestra cómo dividir un objeto en partes y, a continuación, cargar esas partes en un bucket de directorio con el SDK para 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

En los siguientes ejemplos se muestra cómo dividir un objeto en partes y, a continuación, cargar esas partes en un bucket de directorio con el SDK para 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
```

Uso de la AWS CLI

En los siguientes ejemplos se muestra cómo dividir un objeto en partes y, a continuación, cargar esas partes en un bucket de directorio con la AWS CLI. Para usar el comando, sustituya los *marcadores de posición de entrada del usuario* con su propia información.

```
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
```

Para obtener más información, consulte upload-part en la AWS Command Line Interface.

Finalización de una carga multiparte

En los siguientes ejemplos se muestra cómo completar una carga multiparte.

Uso de los SDK de AWS

SDK for Java 2.x

En los siguientes ejemplos se muestra cómo completar una carga multiparte con el SDK para 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

En los siguientes ejemplos se muestra cómo completar una carga multiparte con el SDK para 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}')
```

Uso de la AWS CLI

En este ejemplo se muestra cómo completar una carga multiparte para un bucket de directorio con la AWS CLI. Para usar el comando, sustituya los *marcadores de posición de entrada del usuario* con su propia información.

```
aws s3api complete-multipart-upload --bucket bucket-base-name--azid--x-s3 --
key KEY_NAME --upload-id
 "AS_mgt9RaQE9GEaifATue15dAAAAAAAAAAEMAAAAAAAAADQwNzI4MDU0MjUyMBYAAAAAAAAAAA0AAAAAAAAAAH2AfYAA
 --multipart-upload file://parts.json
```

Este ejemplo utiliza una estructura JSON que describe las partes de la carga multiparte que se deben volver a ensamblar en el archivo completo. En este ejemplo, el prefijo `file://` se usa para cargar la estructura JSON desde un archivo de la carpeta local denominada `parts`.

parts.json:

```
parts.json
{
  "Parts": [
    {
      "ETag": "6b78c4a64dd641a58dac8d9258b88147",
      "PartNumber": 1
    }
  ]
}
```

Para obtener más información, consulte create-multipart-upload en la AWS Command Line Interface.

Anulación de la carga multiparte

En el siguiente ejemplo se muestra cómo anular una carga multiparte.

Uso de los SDK de AWS

SDK for Java 2.x

En el siguiente ejemplo se muestra cómo anular una carga multiparte con el SDK para 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);
            List<MultipartUpload> 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

En el siguiente ejemplo se muestra cómo anular una carga multiparte con el SDK para 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')
```

Uso de la AWS CLI

En el siguiente ejemplo se muestra cómo anular una carga multiparte con la AWS CLI. Para usar el comando, sustituya los *marcadores de posición de entrada del usuario* con su propia información.

```
aws s3api abort-multipart-upload --bucket bucket-base-name--azid--x-s3 --key KEY_NAME
 --upload-id
 "AS_mgt9RaQE9GEaifATue15dAAAAAAAAAAEMAAAAAAAAADQwNzI4MDU0MjUyMBYAAAAAAAAAAA0AAAAAAAAAAH2AfYAA
MAQAAAAB0OxUFeA7LTbWWFS8WYwhrxDxTIDN-pdEEq_agIHqsbg"
```

Para obtener más información, consulte abort-multipart-upload en la AWS Command Line Interface.

Creación de una operación de carga y copia multiparte

En los siguientes ejemplos se muestra cómo copiar objetos de un bucket a otro mediante una carga multiparte.

Uso de los SDK de AWS

SDK for Java 2.x

En el siguiente ejemplo se muestra cómo utilizar una carga multiparte para copiar un objeto mediante programación desde un bucket a otro con el SDK para 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

En el siguiente ejemplo se muestra cómo utilizar una carga multiparte para copiar un objeto mediante programación desde un bucket a otro con el SDK para 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}')
```

Uso de la AWS CLI

En el siguiente ejemplo se muestra cómo utilizar una carga multiparte para copiar un objeto mediante programación desde un bucket a un bucket de directorio con la AWS CLI. Para usar el comando, sustituya los *marcadores de posición de entrada del usuario* con su propia información.

```
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
```

Para obtener más información, consulte upload-part-copy en la AWS Command Line Interface.

Enumeración de las cargas multiparte en curso

Para enumerar las cargas multiparte en curso en un bucket de directorio, puede usar los SDK de AWS o la AWS CLI.

Uso de los SDK de AWS

SDK for Java 2.x

En los siguientes ejemplos se muestra cómo enumerar las cargas multiparte (incompletas) en curso con el SDK para 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

En los siguientes ejemplos se muestra cómo enumerar las cargas multiparte (incompletas) en curso con el SDK para 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}')
```

Uso de la AWS CLI

En los siguientes ejemplos se muestra cómo enumerar las cargas multiparte (incompletas) en curso con la AWS CLI. Para usar el comando, sustituya los *marcadores de posición de entrada del usuario* con su propia información.

```
aws s3api list-multipart-uploads --bucket bucket-base-name--azid--x-s3
```

Para obtener más información, consulte list-multipart-uploads en la AWS Command Line Interface.

Enumeración de las partes de una carga multiparte

En los siguientes ejemplos se muestra cómo enumerar las partes de una carga multiparte en un bucket de directorio.

Uso de los SDK de AWS

SDK for Java 2.x

En los siguientes ejemplos se muestra cómo enumerar las partes de una carga multiparte en un bucket de directorio con el SDK para 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

En los siguientes ejemplos se muestra cómo enumerar las partes de una carga multiparte en un bucket de directorio con el SDK para 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')
```

Uso de la AWS CLI

En los siguientes ejemplos se muestra cómo enumerar las partes de una carga multiparte en un bucket de directorio con la AWS CLI. Para usar el comando, sustituya los *marcadores de posición de entrada del usuario* con su propia información.

```
aws s3api list-parts --bucket bucket-base-name--azid--x-s3 --key KEY_NAME --upload-id
 "AS_mgt9RaQE9GEaifATue15dAAAAAAAAAAEMAAAAAAAAADQwNzI4MDU0MjUyMBYAAAAAAAAAAA0AAAAAAAAAAH2AfYAA
```

Para obtener más información, consulte list-parts en la AWS Command Line Interface.

Copiar un objeto en un bucket de directorio

La operación de copia crea una copia de un objeto que ya esté almacenado en Amazon S3. Puede copiar objetos entre buckets de directorio y buckets de uso general. También puede copiar objetos dentro de un bucket y entre buckets del mismo tipo, por ejemplo, de un bucket de directorio a otro.

Puede crear una copia de un objeto de hasta 5 GB en una única operación atómica. Sin embargo, para copiar un objeto mayor de 5 GB, debe usar las operaciones de la API de carga multiparte. Para obtener más información, consulte Uso de las cargas multiparte con buckets de directorio.

Permisos

Para copiar objetos, debe tener los siguientes permisos:

- Para copiar objetos de un bucket de directorio a otro bucket de directorio, debe disponer del permiso `s3express:CreateSession`.
- Para copiar objetos de buckets de directorio a buckets de uso general, debe tener el permiso `s3express:CreateSession` y el permiso `s3:PutObject` para escribir la copia del objeto en el bucket de destino.
- Para copiar objetos de buckets de uso general en buckets de directorio, debe tener el permiso `s3express:CreateSession` y el permiso `s3:GetObject` para leer el objeto de origen que se está copiando.

  Para obtener más información, consulte CopyObject en la Referencia de la API de Amazon Simple Storage Service.

Cifrado

Amazon S3 cifra automáticamente todos los objetos nuevos que se cargan a un bucket de S3. La configuración de cifrado predeterminada de un bucket de S3 siempre está activada y, como mínimo, se establece en el cifrado del servidor con claves administradas de Amazon S3 (SSE-S3).

Para los buckets de directorio, solo se admite SSE-S3. Para buckets de uso general, puede utilizar SSE-S3 (predeterminado), el cifrado del servidor con claves de AWS Key Management Service (AWS KMS) (SSE-KMS), el cifrado del servidor de doble capa con claves de AWS KMS (DSSE-KMS) o el cifrado del servidor con claves proporcionadas por el cliente (SSE-C).

Si hace una solicitud de copia que establezca los parámetros SSE-C, SSE-KMS o DSSE-KMS en un bucket de directorio como origen o destino, la respuesta devolverá un error.

Etiquetas

Los buckets de directorio no admiten etiquetas. Si copia un objeto que tiene etiquetas de un bucket de uso general a un bucket de directorio, recibirá una respuesta HTTP `501 (Not Implemented)`. Para obtener más información, consulte CopyObject en la Referencia de la API de Amazon Simple Storage Service.

ETags

Las etiquetas de entidad (ETag) para S3 Express One Zone son cadenas alfanuméricas aleatorias, no sumas de comprobación MD5. Para garantizar la integridad del objeto, utilice sumas de comprobación adicionales.

Sumas de comprobación adicionales

S3 Express One Zone le ofrece la opción de elegir el algoritmo de suma de comprobación que se utiliza para validar los datos durante la carga o descarga. Puede seleccionar uno de los siguientes algoritmos de comprobación de integridad de datos Secure Hash Algoritms (SHA) o Cyclic Redundancy Check (CRC): CRC32, CRC32C, SHA-1 y SHA-256. Las sumas de comprobación basadas en MD5 no son compatibles con la clase de almacenamiento S3 Express One Zone.

Para obtener más información, consulte Prácticas recomendadas adicionales para la suma de comprobación de S3.

Características admitidas

Para obtener información sobre las características de Amazon S3 compatibles con S3 Express One Zone, consulte ¿En qué se diferencia S3 Express One Zone?.

Uso de la consola S3 (copia en un bucket de directorio)

Cómo copiar un objeto de un bucket de uso general o de un bucket de directorio a un bucket de uso general

1. Inicie sesión AWS Management Console Management Console y abra la consola de Amazon S3 en https://console.aws.amazon.com/s3/.
2. En el panel de navegación izquierdo, elija Instancias.

3. Elija el bucket desde el que desea copiar objetos:

   - Para copiar desde un bucket de uso general, elija la pestaña Buckets de uso general.
   - Para copiar desde un bucket de directorio, seleccione la pestaña Buckets de directorio.

4. Elija el bucket de uso general o el bucket de directorio que contenga los objetos que desea copiar.
5. Elija la pestaña Objetcts (Objetos). En la página Objetos, seleccione la casilla de verificación situada a la izquierda de los nombres de los objetos que desea copiar.
6. En el menú Actions (Acciones), elija Copy (Copiar).

   Aparece la página Copiar.

7. En Destino, seleccione Bucket de directorio para el tipo de destino. Para especificar la ruta de destino, seleccione Examinar S3, desplácese hasta el destino y seleccione el botón de opción situado a la izquierda del destino. Seleccione Elegir destino en la esquina inferior derecha.

   También puede escribir la ruta de destino.

8. En Sumas de comprobación, elija si desea copiar los objetos con las funciones de suma de comprobación existentes o sustituir las funciones de suma de comprobación existentes por otras nuevas. Al cargar los objetos, tenía la opción de especificar el algoritmo de suma de comprobación que se utilizó para verificar la integridad de los datos. Al copiar el objeto, tiene la opción de elegir una nueva función. Si no ha especificado una suma de comprobación adicional previamente, puede utilizar la sección Sumas de comprobación para agregar una.

   > **Note**
   >
   > Incluso si opta por utilizar la misma función de suma de comprobación, el valor de la suma de comprobación podría cambiar si el objeto tiene un tamaño superior a 16 MB. El valor de la suma de comprobación podría cambiar debido a cómo se calculan las sumas de comprobación para las cargas multiparte. Para obtener más información acerca de cómo puede cambiar la suma de comprobación al copiar el objeto, consulte Uso de sumas de comprobación a nivel de parte para cargas multiparte.

   Para cambiar la función de suma de comprobación, elija Reemplazar por una nueva función de suma de comprobación. Elija la nueva función de suma de comprobación de la lista desplegable. Cuando se copia el objeto, la nueva suma de comprobación se calcula y almacena con el algoritmo especificado.

9. En la esquina inferior derecha, elija Copiar. Amazon S3 copia el objeto en el destino.

Uso de la consola de S3 (copia en un bucket de uso general)

Para copiar un objeto de un bucket de directorio a un bucket de uso general

1. Inicie sesión AWS Management Console Management Console y abra la consola de Amazon S3 en https://console.aws.amazon.com/s3/.
2. En el panel de navegación izquierdo, elija Instancias.
3. Seleccione la pestaña Buckets de directorio.
4. Elija el bucket de directorio que contiene los objetos que desea copiar.
5. Elija la pestaña Objetcts (Objetos). En la página Objetos, seleccione la casilla de verificación situada a la izquierda de los nombres de los objetos que desea copiar.
6. En el menú Actions (Acciones), elija Copy (Copiar).
7. En Destino, seleccione Bucket de uso general para el tipo de destino. Para especificar la ruta de destino, elija Examinar S3, desplácese hasta el destino y seleccione el botón de opción situado a la izquierda del destino. Seleccione Elegir destino en la esquina inferior derecha.

   También puede escribir la ruta de destino.
8. En Sumas de comprobación, elija si desea copiar los objetos con las funciones de suma de comprobación existentes o sustituir las funciones de suma de comprobación existentes por otras nuevas. Al cargar los objetos, tenía la opción de especificar el algoritmo de suma de comprobación que se utilizó para verificar la integridad de los datos. Al copiar el objeto, tiene la opción de elegir una nueva función. Si no ha especificado una suma de comprobación adicional previamente, puede utilizar la sección Sumas de comprobación para agregar una.

> **Note**
>
> Incluso si opta por utilizar la misma función de suma de comprobación, el valor de la suma de comprobación podría cambiar si el objeto tiene un tamaño superior a 16 MB. El valor de la suma de comprobación podría cambiar debido a cómo se calculan las sumas de comprobación para las cargas multiparte. Para obtener más información acerca de cómo puede cambiar la suma de comprobación al copiar el objeto, consulte Uso de sumas de comprobación a nivel de parte para cargas multiparte.

Para cambiar la función de suma de comprobación, elija Reemplazar por una nueva función de suma de comprobación. Elija la nueva función de suma de comprobación de la lista desplegable. Cuando se copia el objeto, la nueva suma de comprobación se calcula y almacena con el algoritmo especificado.

9. En la esquina inferior derecha, elija Copiar. Amazon S3 copia el objeto en el destino.

Uso de los SDK de AWS

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);
      }
}
```

Uso de la AWS CLI

En el siguiente ejemplo de `copy-object` se muestra cómo puede usar la AWS CLI para copiar un objeto de un bucket a otro. Puede copiar objetos entre tipos de buckets. Para ejecutar este comando, sustituya los marcadores de posición de entrada del usuario con su propia información.

```
aws s3api copy-object --copy-source bucket SOURCE_BUCKET/SOURCE_KEY_NAME --
key TARGET_KEY_NAME --bucket TARGET_BUCKET_NAME
```

Para obtener más información, consulte copy-object en la Referencia de los comandos de AWS CLI.

Eliminación de un objeto de un bucket de directorio

Puede eliminar objetos de un bucket de directorio de Amazon S3 mediante la consola de Amazon S3, la AWS Command Line Interface (AWS CLI) o los SDK de AWS. Para obtener más información, consulte Buckets de directorio y ¿Qué es S3 Express One Zone?.

Warning

- La eliminación de un objeto no se puede revertir.
- Esta acción elimina todos los objetos especificados. Al eliminar carpetas, espere a que finalice la acción de eliminación antes de agregar nuevos objetos a la carpeta. De lo contrario, es posible que también se eliminen objetos nuevos.

Note

Cuando elimine varios objetos de un bucket de directorio mediante programación, tenga en cuenta lo siguiente:

- Las claves de objetos de las solicitudes `DeleteObjects` deben contener al menos un carácter que no sea un espacio en blanco. No se admiten cadenas que contengan solo espacios en blanco.
- Las claves de objeto en solicitudes `DeleteObjects` no pueden contener caracteres de control de Unicode, excepto los caracteres de nueva línea (`\n`), tabulador (`\t`) y retorno de carro (`\r`).

Uso de la consola de S3

Para eliminar objetos

1. Inicie sesión en la AWS Management Console y abra la consola de Amazon S3 en https://console.aws.amazon.com/s3/.
2. En el panel de navegación izquierdo, elija Instancias.
3. Seleccione la pestaña Buckets de directorio.
4. Elija el bucket de directorio que contiene los objetos que desea eliminar.
5. Elija la pestaña Objetcts (Objetos). En la lista Objetos, active la casilla de verificación situada a la izquierda del objeto u objetos que desea eliminar.
6. Elija Eliminar.
7. En la página Eliminar objetos, introduzca **`permanently delete`** en el cuadro de texto.
8. Elija Eliminar objetos.

Uso de los SDK de AWS

SDK for Java 2.x

Example

En el siguiente ejemplo se eliminan objetos en un bucket de directorio mediante el 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

En el siguiente ejemplo se eliminan objetos en un bucket de directorio mediante el 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}')
```

Uso de la AWS CLI

En el ejemplo siguiente de `delete-object` se muestra cómo puede utilizar la AWS CLI para eliminar un objeto de un bucket de directorio. Para ejecutar este comando, sustituya los *`user input placeholders`* con su propia información.

```
aws s3api delete-object --bucket bucket-base-name--azid--x-s3 --key KEY_NAME
```

Para obtener más información, consulte delete-object en la Referencia de los comandos de AWS CLI.

Descarga de un objeto en un bucket de directorio

En los siguientes ejemplos de código, se muestra cómo leer datos de un objeto (descargado) en un bucket de directorio de Amazon S3 mediante una operación de la API `GetObject`.

Uso de los SDK de AWS

SDK for Java 2.x

Example

En el siguiente ejemplo de código se muestra cómo leer datos de un objeto en un bucket de directorio mediante 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

En el siguiente ejemplo de código se muestra cómo leer datos de un objeto en un bucket de directorio mediante 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()
```

Uso de la AWS CLI

En el siguiente comando de ejemplo de `get-object`, se muestra cómo puede utilizar la AWS CLI para descargar un objeto de Amazon S3. Este comando obtiene el objeto *KEY_NAME* del

bucket *bucket-base-name*--*azid*--x-s3. El objeto se descargará en un archivo denominado *LOCAL_FILE_NAME*. Para ejecutar este comando, sustituya los *user input placeholders* con su propia información.

```
aws s3api get-object --bucket bucket-base-name--azid--x-s3 --
key KEY_NAME LOCAL_FILE_NAME
```

Para obtener más información, consulte get-object en la Referencia de los comandos de AWS CLI.

Uso de **HeadObject** con buckets de directorio

En los siguientes ejemplos de SDK de AWS y CLI de AWS se muestra cómo usar la operación de la API `HeadObject` para recuperar metadatos de un objeto de un bucket de directorio de Amazon S3 sin devolver el propio objeto.

Uso de los SDK de AWS

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());
```

Uso de la AWS CLI

En el siguiente comando de ejemplo de `head-object`, se muestra cómo puede utilizar la AWS CLI para recuperar metadatos de un objeto. Para ejecutar este comando, sustituya los *`user input placeholders`* con su propia información.

```
aws s3api head-object --bucket bucket-base-name--azid--x-s3 --key KEY_NAME
```

Para obtener más información, consulte head-object en la Referencia de los comandos de AWS CLI.

Seguridad para S3 Express One Zone

La seguridad en la nube de AWS es la mayor prioridad. Como cliente de AWS, se beneficiará de una arquitectura de red y de centros de datos diseñados para satisfacer los requisitos de seguridad de las organizaciones más exigentes. La seguridad es una responsabilidad compartida entre usted y AWS. El modelo de responsabilidad compartida la describe como seguridad de la nube y seguridad en la nube:

- Seguridad de la nube: AWS es responsable de proteger la infraestructura que ejecuta Servicios de AWS en Nube de AWS. Además, AWS proporciona servicios que puede utilizar de forma segura. Auditores externos prueban y verifican periódicamente la eficacia de nuestra seguridad en el marco de los AWS Compliance Programs.

  Para obtener más información acerca de los programas de conformidad que se aplican a Amazon S3 Express One Zone, consulte Servicios de AWS in Scope by Compliance Program.
- Seguridad en la nube: su responsabilidad es determinada por el Servicio de AWS que utilice. También es responsable de otros factores, incluida la confidencialidad de los datos, los requisitos de la empresa y la legislación y los reglamentos aplicables.

Esta documentación le ayudará a comprender cómo aplicar el modelo de responsabilidad compartida cuando se utiliza S3 Express One Zone. En los siguientes temas, se le mostrará cómo configurar S3 Express One Zone a fin de satisfacer sus objetivos de seguridad y conformidad. También aprenderá a usar otros Servicios de AWS que pueden ayudarle a supervisar y proteger sus recursos cuando trabaje con S3 Express One Zone.

Temas

- Protección y cifrado de datos

- AWS Identity and Access Management (IAM) para S3 Express One Zone
- Políticas basadas en identidad de IAM para S3 Express One Zone
- Ejemplos de políticas de bucket de directorio para S3 Express One Zone
- Autorización de CreateSession
- Prácticas recomendadas de seguridad para S3 Express One Zone

Protección y cifrado de datos

Para obtener más información sobre cómo S3 Express One Zone cifra y protege sus datos, consulte los siguientes temas.

Temas

- Cifrado del servidor con claves administradas por Amazon S3 (SSE-S3)
- Cifrado en tránsito
- Sumas de comprobación adicionales
- Eliminación de datos

Cifrado del servidor con claves administradas por Amazon S3 (SSE-S3)

De manera predeterminada, todos los objetos almacenados en el bucket de directorio se almacenan mediante cifrado del servidor con claves de cifrado administradas por Amazon S3 (SSE-S3). No se permiten cargas sin cifrar en buckets de directorio. Para obtener más información, consulte Uso del cifrado del servidor con claves administradas por Amazon S3 (SSE-S3) y Protección de los datos mediante el cifrado.

Los buckets de directorio no admiten el cifrado del servidor con claves de AWS Key Management Service (AWS KMS) (SSE-KMS), el cifrado del servidor de doble capa con claves de AWS Key Management Service (AWS KMS) (DSSE-KMS) o el cifrado del servidor con claves de cifrado proporcionadas por el cliente (SSE-C).

Cifrado en tránsito

Solo se puede acceder a S3 Express One Zone a través de HTTPS (TLS).

S3 Express One Zone utiliza puntos de conexión de API regionales y zonales. Según la operación de API de Amazon S3 que utilice, es necesario un punto de conexión regional o zonal. Puede acceder a los puntos de conexión zonales y regionales a través de un punto de conexión de la nube privada

virtual (VPC) de puerta de enlace. El uso de puntos de enlace de gateway no supone ningún cargo adicional. Para obtener más información sobre los puntos de conexión de las API regionales y zonales, consulte Redes para S3 Express One Zone.

Sumas de comprobación adicionales

S3 Express One Zone le ofrece la opción de elegir el algoritmo de suma de comprobación que se utiliza para validar los datos durante la carga o descarga. Puede seleccionar uno de los siguientes algoritmos de comprobación de integridad de datos Secure Hash Algoritms (SHA) o Cyclic Redundancy Check (CRC): CRC32, CRC32C, SHA-1 y SHA-256. Las sumas de comprobación basadas en MD5 no son compatibles con la clase de almacenamiento S3 Express One Zone.

Para obtener más información, consulte Prácticas recomendadas adicionales para la suma de comprobación de S3.

Eliminación de datos

Puede eliminar uno o más objetos directamente de S3 Express One Zone mediante la consola de Amazon S3, los SDK de AWS, AWS Command Line Interface (AWS CLI) o la API de REST de Amazon S3. Debido a que todos los objetos en los bucket de directorio generan costes de almacenamiento, recomendamos eliminar los objetos cuando ya no los necesite.

Al eliminar un objeto almacenado en un bucket de directorio, también se eliminan de forma recursiva todos los directorios principales en caso de que no contengan ningún objeto que no sea el objeto que se va a eliminar.

> **Note**
>
> S3 Express One Zone no admite la eliminación de la autenticación multifactor (MFA) y el control de versiones de S3.

AWS Identity and Access Management (IAM) para S3 Express One Zone

AWS Identity and Access Management (IAM) es un Servicio de AWS que ayuda a los administradores a controlar de forma segura el acceso a los recursos de AWS. Los administradores de IAM controlan quién se puede autenticar (iniciar sesión) y autorizar (tener permisos) para utilizar los recursos de Amazon S3 en S3 Express One Zone. El uso de IAM no está sujeto a ningún cargo adicional.

De forma predeterminada, los usuarios no tienen permisos para los buckets de directorio ni para las operaciones de S3 Express One Zone. Para conceder permisos de acceso a los buckets de directorio, puede usar IAM para crear usuarios, grupos o roles y asociar permisos a esas identidades. Para obtener más información sobre IAM, consulte la sección Prácticas recomendadas de IAM en la Guía del usuario de IAM;.

Para proporcionar acceso, puede agregar permisos a sus usuarios, grupos o roles a través de los siguientes medios:

- Usuarios y grupos en AWS IAM Identity Center: Cree un conjunto de permisos. Siga las instrucciones de Creación de un conjunto de permisos en la Guía del usuario de AWS IAM Identity Center.
- Usuarios administrados en IAM a través de un proveedor de identidades: Cree un rol para la federación de identidades. Siga las instrucciones de Creación de un rol para un proveedor de identidades de terceros (federación) en la Guía del usuario de IAM.
- Usuarios y roles de IAM: Cree un rol que el usuario pueda asumir. Siga las instrucciones de Creación de un rol para delegar permisos a un usuario de IAM de la Guía del usuario de IAM.

De forma predeterminada, los buckets de directorio son privados y solo pueden acceder a ellos los usuarios a los que se concede explícitamente el acceso. El límite de control de acceso para buckets de directorio se establece únicamente en el nivel de bucket. Por el contrario, el límite de control de acceso para los buckets de uso general se puede establecer en el nivel de bucket, prefijo o etiqueta de objeto. Esta diferencia significa que los buckets de directorio son el único recurso que puede incluir en las políticas de bucket o en las políticas de identidad de IAM para el acceso a S3 Express One Zone.

Con S3 Express One Zone, además de la autorización de IAM, autentica y autoriza las solicitudes mediante un nuevo mecanismo basado en sesiones que se controla mediante la operación de la API `CreateSession`. Puede utilizar `CreateSession` para solicitar credenciales temporales que otorgan acceso de baja latencia a su bucket. Estas credenciales temporales se asignan a un bucket de directorio específico.

Para trabajar con `CreateSession`, le recomendamos que utilice la última versión de los SDK de AWS o que utilice la AWS Command Line Interface (AWS CLI). Los SDK de AWS compatibles con AWS CLI controlan el establecimiento, la actualización y la finalización de la sesión en su nombre.

Utiliza tokens de sesión únicamente con operaciones zonales (de nivel de objeto) (excepto para `CopyObject` y`HeadBucket`) para distribuir la latencia asociada a la autorización entre varias

solicitudes de una sesión. Para las operaciones de la API de puntos de conexión regionales (operaciones de nivel de bucket), se utiliza la autorización de IAM, que no implica la administración de una sesión. Para obtener más información, consulte AWS Identity and Access Management (IAM) para S3 Express One Zone y Autorización de `CreateSession`.

Para obtener más información acerca de IAM para S3 Express One Zone, consulte los siguientes temas.

Temas

Entidades principales

Cuando crea una política basada en recursos para conceder acceso a sus buckets, debe utilizar el elemento `Principal` para especificar la persona o aplicación que puede realizar una solicitud para realizar una acción o una operación en ese recurso. Para las políticas de buckets de directorio, puede utilizar los siguientes principios:

- Una cuenta de AWS.
- Un usuario de IAM
- Un rol de IAM
- Un usuario federado

Para obtener más información, consulte Principal en la Guía del usuario de IAM.

Recursos

Los nombres de recursos de Amazon (ARN) para los buckets de directorio contienen el espacio de nombres `s3express`, la Región de AWS, el ID de cuenta de AWS y el nombre del bucket de directorio, que incluye el ID de zona de disponibilidad. Para acceder y realizar acciones en el bucket de directorio, debe utilizar el formato de ARN siguiente:

```
arn:aws:s3express:region:account-id:bucket/base-bucket-name--azid--x-s3
```

Para obtener más información, consulte Amazon Resource Names (ARNs) en la Guía del usuario de IAM. Para obtener más información sobre los recursos, consulte Elementos de política JSON de IAM: Resource en la Guía del usuario de IAM.

Acciones para S3 Express One Zone

En una política basada en identidad de IAM o una política basada en recursos, usted define qué acciones de S3 se permiten o deniegan. Las acciones de S3 Express One Zone corresponden a operaciones de la API concretas. S3 Express One Zone tiene un espacio de nombres de IAM único que es distinto del espacio de nombres estándar de Amazon S3. Este espacio de nombres es `s3express`.

Al conceder el permiso `s3express:CreateSession`, la operación de la API `CreateSession` puede recuperar los tokens de sesión al acceder a las operaciones de la API de puntos de conexión zonales (o de nivel de objeto). Estos tokens de sesión devuelven las credenciales que se utilizan para conceder acceso a todas las demás operaciones de la API de puntos de conexión zonales. Como resultado, no es necesario conceder permisos de acceso a las operaciones de API zonales mediante políticas de IAM. En su lugar, el token de sesión permite el acceso.

Para obtener más información sobre las operaciones de API de puntos de conexión zonales y regionales, consulte Redes para S3 Express One Zone. Para obtener más información sobre la operación de la API `CreateSession`, consulte CreateSession en la Referencia de la API de Amazon Simple Storage Service.

Puede especificar las siguientes acciones en el elemento `Action` de una declaración de política de IAM. Utilice políticas para conceder permisos para realizar una operación en AWS. Cuando utiliza una acción en una política, normalmente permite o deniega el acceso a la operación de la API con el mismo nombre. No obstante, en algunos casos, una sola acción controla el acceso a más de una operación de la API. El acceso a las acciones en el nivel de bucket solo se puede conceder en las políticas basadas en identidad de IAM (usuario o rol), no en las políticas de bucket.

Acciones y claves de condición para S3 Express One Zone

Acción	API	Descripción	Nivel de acceso	Claves de condición
`s3express:CreateBucket`	`CreateBucket`	Concede permiso para crear un nuevo bucket.	Escritura	`s3express:authType` `s3express:LocationName` `s3express:ResourceAccount` `s3express:signatureversion` `s3express:TlsVersion` `s3express:x-amz-content-sha256`
`s3express:CreateSession`	`CreateSession`	Concede permiso para crear un token de sesión, que se utiliza para conceder acceso a todas las operaciones de la API zonal (en el nivel de objeto), como, `PutObject` , `GetObject` , etc.	Escritura	`s3express:authType` `s3express:SessionMode` `s3express:ResourceAccount`

Acción	API	Descripción	Nivel de acceso	Claves de condición
				`s3express:signatureversion` `s3express:signatureAge` `s3express:TlsVersion` `s3express:x-amz-content-sha256`

Acción	API	Descripción	Nivel de acceso	Claves de condición
`s3express:DeleteBucket`	`DeleteBucket`	Concede permiso para eliminar el bucket nombrado en el URI.	Escritura	`s3express:authType` `s3express:ResourceAccount` `s3express:signatureversion` `s3express:TlsVersion` `s3express:x-amz-content-sha256`

Acción	API	Descripción	Nivel de acceso	Claves de condición
`s3express:DeleteBucketPolicy`	`DeleteBucketPolicy`	Concede permiso para eliminar la política en un bucket especificado.	Administración de permisos	`s3express:authType` `s3express:ResourceAccount` `s3express:signatureversion` `s3express:TlsVersion` `s3express:x-amz-content-sha256`

Acción	API	Descripción	Nivel de acceso	Claves de condición
s3express:GetBucketPolicy	GetBucketPolicy	Concede permiso para devolver la política del bucket especificado.	Leer	s3express:authType s3express:ResourceAccount s3express:signatureversion s3express:TlsVersion s3express:x-amz-content-sha256

Acción	API	Descripción	Nivel de acceso	Claves de condición
`s3express:ListAllMyDirectoryBuckets`	`ListDirectoryBuckets`	Concede permiso para enumerar todos los buckets de directorio propiedad del remitente autenticado de la solicitud.	Enumeración	`s3express:authType` `s3express:ResourceAccount` `s3express:signatureversion` `s3express:TlsVersion` `s3express:x-amz-content-sha256`

Acción	API	Descripción	Nivel de acceso	Claves de condición
`s3express:PutBucketPolicy`	`PutBucketPolicy`	Concede permiso para agregar o reemplazar una política de bucket.	Administración de permisos	`s3express:authType` `s3express:ResourceAccount` `s3express:signatureversion` `s3express:TlsVersion` `s3express:x-amz-content-sha256`

Claves de condición de S3 Express One Zone

S3 Express One Zone define las siguientes claves de condición que se pueden utilizar en el elemento `Condition` de una política de IAM. Puede utilizar estas claves para ajustar más las condiciones en las que se aplica la instrucción de política.

Clave de condición	Descripción	Tipo
`s3express:authType`	Filtra el acceso por método de autenticación. Para restringir las solicitudes entrantes para que usen un método de autenticación específico, puede usar esta clave de condición opcional. Por ejemplo, puede utilizar esta clave de	Cadena

Clave de condición	Descripción	Tipo
	condición para permitir solo el encabezado `Authorization` de HTTP que se utilizará en la autenticación de solicitudes. Valores válidos: `REST-HEADER` , `REST-QUERY-STRING`	
`s3express:LocationName`	Filtra el acceso a la operación de la API `CreateBucket` mediante un ID de zona de disponibilidad (ID de AZ) específico, por ejemplo, `usw2-az1`. Ejemplo de valor: `usw2-az1`	Cadena
`s3express:ResourceAccount`	Filtra el acceso por el ID de la Cuenta de AWS del propietario del recurso. Para restringir el acceso de usuarios, roles o aplicaciones a los buckets de directorio que pertenecen a un ID de Cuenta de AWS específico, puede utilizar la clave de condición de `aws:ResourceAccount` o `s3express:ResourceAccount` . Puede usar esta clave de condición en las políticas de identidad de AWS Identity and Access Management (IAM) o en las políticas de puntos de conexión de nube privada virtual (VPC). Por ejemplo, puede usar esta clave de condición para restringir que los clientes dentro de su VPC accedan a los buckets que no posee. Ejemplo de valor: `111122223333`	Cadena

Clave de condición	Descripción	Tipo
`s3express:SessionMode`	Filtra el acceso según el permiso solicitado por la operación de la API `CreateSession` . De forma predeterminada, la sesión es `ReadWrite` . Puede utilizar esta clave de condición para limitar el acceso a `ReadOnly` o denegar explícitamente el acceso a `ReadWrite` . Para obtener más información, consulte Ejemplos de políticas de bucket de directorio para S3 Express One Zone y CreateSession en la Referencia de la API de Amazon Simple Storage Service. Valores válidos: `ReadWrite` , `ReadOnly`	Cadena
`s3express:signatureAge`	Filtra el acceso por la antigüedad en milisegundos de la firma de la solicitud. Esta condición solo funciona para las direcciones URL prefirmadas. En AWS Signature Version 4, la clave de firma es válida por un plazo máximo de siete días. Por lo tanto, las firmas también son válidas por un plazo máximo de siete días. Para obtener más información, consulte Introducción a la firma de solicitudes en la Referencia de la API de Amazon Simple Storage Service. Puede usar esta condición para limitar aún más la antigüedad de la firma. Ejemplo de valor: `600000`	Numérico

Clave de condición	Descripción	Tipo
`s3express:signaturversion`	Identifica la versión de AWS Signature que desea admitir con las solicitudes autenticadas. Para las solicitudes autenticadas, S3 Express One Zone admite Signature Version 4. Valores válidos: `"AWS4-HMAC-SHA256"` (identifica Signature Version 4)	Cadena
`s3express:TlsVersion`	Filtra el acceso por la versión de TLS que utiliza el cliente Puede utilizar la clave de condición `s3:TlsVersion` para escribir políticas de bucket, de punto de conexión de nube privada virtual (VPCE) o de IAM que restrinjan el acceso de aplicaciones o usuarios a buckets de directori o en función de la versión de TLS que utilice el cliente. También puede utilizar esta clave de condición para escribir políticas que requieran una versión mínima de TLS. Ejemplo de valor: `1.3`	Numérico

Clave de condición	Descripción	Tipo
`s3express:x-amz-content-sha256`	Filtra el acceso en función del contenido sin firmar en el bucket. Puede utilizar esta clave de condición para no permitir el contenido sin firmar en su bucket. Cuando se utiliza Signature Version 4, para las solicitudes que utilizan el encabezado `Authorization` , agregue el encabezado `x-amz-content-sha256` en el cálculo de la firma y luego establezca su valor en la carga de hash. Puede usar esta clave de condición en su política de bucket para denegar cualquier carga en la que las cargas no estén firmadas. Por ejemplo: • Denegar las cargas que usen el encabezado `Authorization` para autenticar las solicitudes, pero no firmar la carga. Para obtener más información, consulte Transferencia de carga en un solo fragmento en la Referencia de la API de Amazon Simple Storage Service. • Denegar las cargas que utilicen URL prefirmadas. Las URL prefirmadas siempre tienen una `UNSIGNED_PAYLOAD` . Para obtener más información, consulte Autenticación de solicitudes y Métodos de autenticación en la Referencia de la API de Amazon Simple Storage Service. Valor válido: `UNSIGNED-PAYLOAD`	Cadena

Cómo se autorizan y autentican las operaciones de la API

En la siguiente tabla se muestra la información de autorización y autenticación para las operaciones de la API de S3 Express One Zone. Para cada operación de la API, la tabla muestra el nombre de la operación de la API, la acción de IAM, el tipo de punto de conexión (regional o zonal) y el mecanismo de autorización (basado en IAM o en sesiones). En esta tabla también se indica dónde se admite el acceso entre cuentas. El acceso a las acciones en el nivel de bucket solo se puede conceder en las políticas basadas en identidad de IAM (usuario o rol), no en las políticas de bucket.

API	Tipo de punto de conexión	Acción de IAM	Acceso entre cuentas
`CreateBucket`	Regional	`s3express:CreateBucket`	No
`DeleteBucket`	Regional	`s3express:DeleteBucket`	No
`ListDirectoryBuckets`	Regional	`s3express:ListAllMyDirectoryBuckets`	No
`PutBucketPolicy`	Regional	`s3express:PutBucketPolicy`	No
`GetBucketPolicy`	Regional	`s3express:GetBucketPolicy`	No
`DeleteBucketPolicy`	Regional	`s3express:DeleteBucketPolicy`	No
`CreateSession`	Zonal	`s3express:CreateSession`	Sí
`CopyObject`	Zonal	`s3express:CreateSession`	Sí
`DeleteObject`	Zonal	`s3express:CreateSession`	Sí
`DeleteObjects`	Zonal	`s3express:CreateSession`	Sí
`HeadObject`	Zonal	`s3express:CreateSession`	Sí
`PutObject`	Zonal	`s3express:CreateSession`	Sí

API	Tipo de punto de conexión	Acción de IAM	Acceso entre cuentas
`GetObjectAttributes`	Zonal	`s3express:CreateSession`	Sí
`ListObjectsV2`	Zonal	`s3express:CreateSession`	Sí
`HeadBucket`	Zonal	`s3express:CreateSession`	Sí
`CreateMultipartUpload`	Zonal	`s3express:CreateSession`	Sí
`UploadPart`	Zonal	`s3express:CreateSession`	Sí
`UploadPartCopy`	Zonal	`s3express:CreateSession`	Sí
`CompleteMultipartUpload`	Zonal	`s3express:CreateSession`	Sí
`AbortMultipartUpload`	Zonal	`s3express:CreateSession`	Sí
`ListParts`	Zonal	`s3express:CreateSession`	Sí
`ListMultipartUploads`	Zonal	`s3express:CreateSession`	Sí

Políticas basadas en identidad de IAM para S3 Express One Zone

Antes de poder crear buckets de directorio o utilizar la clase de almacenamiento Amazon S3 Express One Zone, debe conceder los permisos necesarios a su rol o usuarios de AWS Identity and Access Management (IAM). Esta política de ejemplo permite el acceso a la operación de la API `CreateSession` (para utilizarla con las operaciones de la API del punto de conexión zonal [de nivel de objeto]) y a todas las operaciones de la API del punto de conexión regional (de nivel de bucket). Esta política permite que la operación de la API `CreateSession` se utilice con todos los buckets

de directorio, pero las operaciones de la API de punto de conexión regional solo se permiten con el bucket de directorio especificado. Para utilizar esta política de ejemplo, sustituya *user input placeholders* por su propia información.

```
{
     "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": "*"
         }
     ]
 }
```

Ejemplos de políticas de bucket de directorio para S3 Express One Zone

En esta sección se proporcionan ejemplos de políticas de bucket de directorio para utilizarlas con la clase de almacenamiento Amazon S3 Express One Zone. Para utilizar estas políticas, sustituya *user input placeholders* por su información.

El siguiente ejemplo de política de bucket permite que el ID de Cuenta de AWS *111122223333* utilice la operación de la API `CreateSession` con la sesión `ReadWrite` predeterminada para el bucket de directorio especificado. Esta política otorga acceso a las operaciones de la API del punto de conexión zonal (de nivel de objeto).

Example – Política de buckets para permitir llamadas **`CreateSession`** con la sesión **`ReadWrite`** predeterminada

```
{
    "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 – Política de buckets para permitir llamadas **`CreateSession`** con una sesión **`ReadOnly`**

El siguiente ejemplo de política de bucket permite que el ID de Cuenta de AWS *111122223333* utilice la operación de la API `CreateSession`. Esta política utiliza la clave de condición `s3express:SessionMode` con el valor `ReadOnly` para establecer una sesión de solo lectura.

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Sid": "ReadOnlyAccess",
            "Effect": "Allow",
            "Principal": {
                "AWS": "111122223333"
            },
            "Action": "s3express:CreateSession",
```

```
            "Resource": "*",
            "Condition": {
                "StringEquals": {
                    "s3express:SessionMode": "ReadOnly"
                }
            }
        }
    ]
}
```

Example – Política de bucket para permitir el acceso entre cuentas para llamadas **`CreateSession`**

El siguiente ejemplo de política de bucket permite que el ID de Cuenta de AWS *111122223333* utilice la operación de la API `CreateSession` para el bucket de directorio especificado que es propiedad del ID de Cuenta de AWS *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"

        }
    ]
}
```

Autorización de **`CreateSession`**

Amazon S3 Express One Zone admite tanto la autorización de AWS Identity and Access Management (IAM de AWS) como la autorización basada en sesiones:

- Para utilizar las operaciones de la API de puntos de conexión regionales (operaciones de nivel de bucket o plano de control) con S3 Express One Zone, utilice el modelo de autorización de IAM, que no implica la administración de sesiones. Los permisos se conceden para las acciones de forma individual. Para obtener más información, consulte AWS Identity and Access Management (IAM) para S3 Express One Zone.
- Para utilizar las operaciones de la API de puntos de conexión zonales (operaciones de nivel de objeto o plano de datos), utilice la operación de la API `CreateSession` para crear y administrar sesiones optimizadas para la autorización de solicitudes de datos con baja latencia. Para recuperar y usar un token de sesión, debe permitir la acción `s3express:CreateSession` para su bucket de directorio en una política basada en identidades o en una política de bucket. Para obtener más información, consulte AWS Identity and Access Management (IAM) para S3 Express One Zone. Si accede a S3 Express One Zone en la consola de Amazon S3, a través de AWS Command Line Interface (AWS CLI) o mediante los SDK de AWS, S3 Express One Zone crea una sesión en su nombre.

Si utiliza la API de REST de Amazon S3, podrá utilizar la operación de la API `CreateSession` para obtener credenciales de seguridad temporales que incluyan un ID de clave de acceso, una clave de acceso secreta, un token de sesión y una fecha de caducidad. Las credenciales temporales proporcionan los mismos permisos que las credenciales de seguridad a largo plazo, como las credenciales de usuario de IAM, pero deben incluir un token de sesión.

Modo Sesión

El modo Sesión define el alcance de la sesión. En su política de bucket, puede especificar la clave de condición `s3express:SessionMode` para controlar quién puede crear una sesión `ReadWrite` o `ReadOnly`. Para obtener más información sobre las sesiones `ReadWrite` o `ReadOnly`, consulte el parámetro `x-amz-create-session-mode` para CreateSession en la Referencia de la API de Amazon S3. Para obtener información acerca de la creación de una política de bucket, consulte Ejemplos de políticas de bucket de directorio para S3 Express One Zone.

Token de sesión

Cuando realice una llamada utilizando las credenciales de seguridad temporales, la llamada debe incluir un token de sesión. El token de sesión se devuelve junto con las credenciales temporales. El token de sesión se asigna a su bucket de directorio y se utiliza para comprobar que las credenciales de seguridad son válidas y no han caducado. Para proteger sus sesiones, las credenciales de seguridad temporales caducan a los 5 minutos.

`CopyObject` y `HeadBucket`

Las credenciales de seguridad temporales se asignan a un bucket de directorio específico y se habilitan automáticamente para todas las llamadas a la API de operaciones zonales (de nivel de objeto) en un bucket de directorio determinado. A diferencia de otras operaciones de la API de puntos de conexión zonales, `CopyObject` y `HeadBucket` no utilizan la autenticación `CreateSession`. Todas las solicitudes `CopyObject` y `HeadBucket` deben autenticarse y firmarse con credenciales de IAM. Sin embargo, `CopyObject` y `HeadBucket` siguen estando autorizadas por `s3express:CreateSession`, al igual que otras operaciones de la API de puntos de conexión zonales.

Para obtener más información, consulte CreateSession en la Referencia de la API de Amazon Simple Storage Service.

Prácticas recomendadas de seguridad para S3 Express One Zone

Amazon S3 Express One Zone proporciona una serie de características de seguridad que debe tener en cuenta a la hora de desarrollar e implementar sus propias políticas de seguridad. Las siguientes prácticas recomendadas son directrices generales y no suponen una solución de seguridad completa. Puesto que es posible que estas prácticas recomendadas no sean adecuadas o suficientes para su entorno, considérelas como recomendaciones útiles en lugar de como normas.

Configuración predeterminada de Bloqueo de acceso público y Propiedad de objetos

Para usar la clase de almacenamiento S3 Express One Zone, debe usar un bucket de directorio de S3. Los buckets de directorio admiten S3 Block Public Access y S3 Object Ownership. Estas características de S3 se utilizan para auditar y administrar el acceso a sus buckets y objetos.

De forma predeterminada, todas las configuraciones de Bloqueo de acceso público estarán activas para los buckets de directorio. Además, la Propiedad de objetos está configurada como aplicada al propietario del bucket, lo que significa que las listas de control de acceso (ACL) están deshabilitadas. Esta configuración no se puede modificar. Para obtener más información sobre el uso de estas características, consulte the section called “Bloquear acceso público” y the section called “Control de la propiedad de objetos”.

> **Note**
>
> No puede conceder acceso a objetos almacenados en buckets de directorio. Solo puede conceder acceso a sus buckets de directorio. El modelo de autorización de S3 Express One

Zone es diferente del modelo de autorización de Amazon S3. Para obtener más información, consulte Autorización de `CreateSession`.

Autenticación y autorización

Los mecanismos de autenticación y autorización de S3 Express One Zone varían en función de si realiza solicitudes a las operaciones de la API de puntos de conexión zonales o a las operaciones de la API de puntos de conexión regionales. Las operaciones de la API zonales son operaciones de nivel de objeto (plano de datos). Las operaciones de la API regionales son operaciones de nivel de bucket (plano de control).

Con S3 Express One Zone, autentica y autoriza las solicitudes a las operaciones de la API de puntos de conexión zonales mediante un nuevo mecanismo basado en sesiones que está optimizado para brindar la latencia más baja. Con la autenticación basada en sesiones, los SDK de AWS utilizan la operación de la operación de la API `CreateSession`para solicitar credenciales temporales que otorgan acceso de baja latencia a su bucket de directorio. Estas credenciales temporales se asignan a un bucket de directorio específico y caducan a los 5 minutos. Puede usar estas credenciales temporales para firmar llamadas a la API zonal (de nivel de objeto). Para obtener más información, consulte Autorización de `CreateSession`.

Firma de solicitudes con las credenciales de S3 Express One Zone

Utiliza sus credenciales de S3 Express One Zone para firmar las solicitudes de API de punto de conexión zonal (de nivel de objeto) con AWS Signature Version 4, con `s3express` como el nombre del servicio. Al firmar las solicitudes, utilice la clave secreta que se devuelve de `CreateSession` y proporcione también el token de sesión junto con `x-amzn-s3session-token header`. Para obtener más información, consulte CreateSession.

Los SDK de AWS compatibles con la clase S3 Express One Zone administran las credenciales y la firma en su nombre. Le recomendamos que utilice los SDK de AWS para S3 Express One Zone a fin de actualizar las credenciales y firmar las solicitudes por usted.

Firma de las solicitudes con credenciales de IAM

Todas las llamadas a la API regionales (de nivel de bucket) deben autenticarse y firmarse con credenciales de AWS Identity and Access Management (IAM) en lugar de con credenciales de sesión temporales. Las credenciales de IAM se componen del ID de clave de acceso y la clave de acceso

secreta de las identidades de IAM. Todas las solicitudes `CopyObject` y `HeadBucket` también deben autenticarse y firmarse con credenciales de IAM.

Para lograr la latencia más baja en sus llamadas de operaciones zonales (de nivel de objeto), recomendamos utilizar las credenciales de S3 Express One Zone obtenidas al llamar a `CreateSession` para firmar sus solicitudes, excepto las solicitudes dirigidas a `CopyObject` y `HeadBucket`.

Utilizar AWS CloudTrail

AWS CloudTrail proporciona un registro de las medidas adoptadas por un usuario, un rol o un Servicio de AWS en Amazon S3. Puede utilizar la información recopilada por CloudTrail para determinar lo siguiente:

- La solicitud que se realizó a Amazon S3
- La dirección IP desde la que se realizó la solicitud
- Quién realizó la solicitud
- La hora a la que se realizó la solicitud
- Detalles adicionales sobre la solicitud

Cuando se configura una Cuenta de AWS, los eventos de administración de CloudTrail se activan de forma predeterminada. Las siguientes operaciones de la API de puntos de conexión regionales (operaciones de API de nivel de bucket o plano de control) se registran en CloudTrail.

- CreateBucket
- DeleteBucket
- DeleteBucketPolicy
- PutBucketPolicy
- GetBucketPolicy
- ListDirectoryBuckets
- ListMultipartUploads

> **Note**
>
> `ListMultipartUploads` es una operación de la API de puntos de conexión zonales. Sin embargo, se registra en CloudTrail como un evento de administración. Para obtener más

información, consulte ListMultipartUploads en la Referencia de la API de Amazon Simple Storage Service.

De forma predeterminada, los registros de seguimiento de CloudTrail no registran eventos de datos, pero pueden configurarse para los buckets de directorio que usted especifique o para que registren eventos de datos para todos los buckets de directorio incluidos en la cuenta de AWS. Las siguientes operaciones de la API de puntos de conexión zonales (operaciones de API de objeto o plano de datos) se registran en CloudTrail.

- AbortMultipartUpload
- CompleteMultipartUpload
- CreateSession
- CopyObject
- CreateMultipartUpload
- DeleteObject
- DeleteObjects
- GetObject
- GetObjectAttributes
- HeadBucket
- HeadObject
- ListObjectsV2
- ListParts
- PutObject
- UploadPart
- UploadPartCopy

Para obtener información sobre el uso de AWS CloudTrail con S3 Express One Zone, consulte Registro con AWS CloudTrail para S3 Express One Zone.

Implementación de la monitorización mediante las herramientas de supervisión de AWS

La monitorización es una parte importante del mantenimiento de la fiabilidad, la seguridad, la disponibilidad y el rendimiento de Amazon S3 y las soluciones de AWS. AWS brinda herramientas

y servicios para ayudarlo a monitorizar Amazon S3 y los otros servicios de Servicios de AWS. Por ejemplo, puede monitorizar métricas de Amazon CloudWatch para Amazon S3, concretamente las métricas de almacenamiento `BucketSizeBytes` y `NumberOfObjects`.

Los objetos almacenados en la clase de almacenamiento S3 Express One Zone no se reflejarán en las métricas de almacenamiento `BucketSizeBytes` y `NumberOfObjects` para Amazon S3. No obstante, las métricas de almacenamiento `BucketSizeBytes` y `NumberOfObjects` son compatibles con S3 Express One Zone. Para ver las métricas de su elección, puede diferenciar entre las clases de almacenamiento de Amazon S3 y la clase de almacenamiento S3 Express One Zone especificando una dimensión `StorageType`. Para obtener más información, consulte Monitorización de métricas con Amazon CloudWatch.

Para obtener más información, consulte Monitorización de métricas con Amazon CloudWatch y Monitorización de Amazon S3.

Registro con AWS CloudTrail para S3 Express One Zone

Amazon S3 está integrado con AWS CloudTrail, un servicio que proporciona un registro de las acciones que realiza un usuario, un rol o un servicio de AWS. CloudTrail captura todas las llamadas a la API para Amazon S3 como eventos. Mediante la información recopilada por CloudTrail, puede determinar la solicitud que se realizó a Amazon S3, la dirección IP desde la que se realizó, cuándo se realizó y detalles adicionales. Cuando se produce una actividad de evento admitida en Amazon S3, dicha actividad se registra en un evento de CloudTrail. Puede usar un registro de seguimiento de AWS CloudTrail para registrar eventos de administración y eventos de datos para S3 Express One Zone. Para obtener más información, consulte Eventos de Amazon S3 CloudTrail y What is AWS CloudTrail? en la Guía del usuario de AWS CloudTrail.

Eventos de administración de CloudTrail para S3 Express One Zone

De forma predeterminada, CloudTrail registra acciones en el nivel de bucket para buckets de directorio como eventos de administración. `eventsource` para eventos de administración de CloudTrail para S3 Express One Zone es `s3express.amazonaws.com`. Cuando se configura una cuenta de AWS, los eventos de administración de CloudTrail se activan de forma predeterminada. Las siguientes operaciones de la API de puntos de conexión regionales (operaciones de API de nivel de bucket o plano de control) se registran en CloudTrail.

- `CreateBucket`
- `DeleteBucket`

- DeleteBucketPolicy
- PutBucketPolicy
- GetBucketPolicy
- ListDirectoryBuckets
- ListMultipartUploads

> **Note**
>
> `ListMultipartUploads` es una operación de la API de puntos de conexión zonales. Sin embargo, esta operación de la API se registra en CloudTrail como un evento de administración. Para obtener más información, consulte `ListMultipartUploads` en la Referencia de la API de Amazon Simple Storage Service.

Para obtener más información sobre los eventos de administración de CloudTrail, consulte Logging management events en la Guía del usuario de AWS CloudTrail.

Eventos de datos de CloudTrail para S3 Express One Zone

Los eventos de datos proporcionan información sobre las operaciones de recursos realizadas en o dentro de un recurso (por ejemplo, leer o escribir en un objeto de Amazon S3). Se denominan también operaciones del plano de datos. Los eventos de datos suelen ser actividades de gran volumen. De forma predeterminada, los registros de seguimiento de AWS CloudTrail no registran eventos de datos, pero pueden configurarse para que registren eventos de datos de objetos almacenados en buckets de uso general y de directorio. Para obtener más información, consulte Habilitar el registro de objetos en un bucket mediante la consola.

Cuando registra los eventos de datos de un registro de seguimiento en CloudTrail, tiene la opción de elegir usar selectores de eventos avanzados o selectores de eventos básicos. Para registrar eventos de datos de objetos almacenados en buckets de directorio, debe utilizar selectores de eventos avanzados. Al configurar los selectores de recursos avanzados, elegirá o especificará el tipo de recurso para S3 Express One Zone, que es `AWS::S3Express::Object`.

Las siguientes operaciones de la API de puntos de conexión zonales (operaciones de API de objeto o plano de datos) se registran en CloudTrail.

- AbortMultipartUpload

- CompleteMultipartUpload
- CreateSession
- CopyObject
- CreateMultipartUpload
- DeleteObject
- DeleteObjects
- GetObject
- GetObjectAttributes
- HeadBucket
- HeadObject
- ListObjectsV2
- ListParts
- PutObject
- UploadPart
- UploadPartCopy

Para obtener más información sobre los eventos de administración de CloudTrail, consulte Logging data events en la Guía del usuario de AWS CloudTrail.

Para obtener más información acerca de los eventos de CloudTrail para S3 Express One Zone, consulte los siguientes temas:

Temas

- Ejemplos de archivos de registro de CloudTrail para S3 Express One Zone

Ejemplos de archivos de registro de CloudTrail para S3 Express One Zone

Un registro de CloudTrail incluye información acerca de la operación de la API solicitada, la fecha y la hora de la operación, los parámetros de la solicitud, etc. En este tema se incluyen ejemplos de eventos de datos de CloudTrail y eventos de administración para S3 Express One Zone.

Temas

- Ejemplos de archivos de registro de eventos de datos de CloudTrail para Amazon S3 Express One Zone

Ejemplos de archivos de registro de eventos de datos de CloudTrail para Amazon S3 Express One Zone

En el siguiente ejemplo, se muestra un archivo de registro de CloudTrail que demuestra CreateSession.


```
    {
        "eventVersion": "1.09",
        "userIdentity": {
          "type": "AssumedRole",
          "principalId": "AROAIDPPEZS35WEXAMPLE:AssumedRoleSessionName",
          "arn": "arn:aws:sts::111122223333assumed-role/RoleToBeAssumed/MySessionName",
          "accountId": "111122223333",
          "accessKeyId": "AKIAIOSFODNN7EXAMPLE",
          "sessionContext": {
            "sessionIssuer": {
              "type": "Role",
              "principalId": "AROAIDPPEZS35WEXAMPLE",
              "arn": "arn:aws:iam::111122223333:role/RoleToBeAssumed",
              "accountId": "111122223333",
              "userName":"RoleToBeAssumed
            },

            "attributes": {
              "creationDate": "2024-07-02T00:21:16Z",
            "mfaAuthenticated": "false"
            }
          }
        },
        "eventTime": "2024-07-02T00:22:11Z",
        "eventSource": "s3express.amazonaws.com",
        "eventName": "CreateSession",
        "awsRegion": "us-west-2",
        "sourceIPAddress": "72.21.198.68",
        "userAgent": "aws-sdk-java/2.20.160-SNAPSHOT
 Linux/5.10.216-225.855.amzn2.x86_64 OpenJDK_64-Bit_Server_VM/11.0.23+9-LTS
 Java/11.0.23 vendor/Amazon.com_Inc. md/internal exec-env/AWS_Lambda_java11 io/sync
 http/Apache cfg/retry-mode/standard",
        "requestParameters": {
          "bucketName": "bucket-base-name--usw2-az1--x-s3".
            "host": "bucket-base-name--usw2-az1--x-s3.s3express-usw2-az1.us-
west-2.amazonaws.com",
            "x-amz-create-session-mode": "ReadWrite"
```

```
        },
        "responseElements": {
            "credentials": {
                "accessKeyId": "AKIAI44QH8DHBEXAMPLE"
                "expiration": ""Mar 20, 2024, 11:16:09 PM",
                "sessionToken": "<session token string>"
           },
        },
        "additionalEventData": {
            "SignatureVersion": "SigV4",
            "cipherSuite": "TLS_AES_128_GCM_SHA256",
            "bytesTransferredIn": 0,
            "AuthenticationMethod": "AuthHeader",
            "xAmzId2": "q6xhNJYmhg",
            "bytesTransferredOut": 1815,
            "availabilityZone": "usw2-az1"
          },
          "requestID": "28d2faaf-3319-4649-998d-EXAMPLE72818",
          "eventID": "694d604a-d190-4470-8dd1-EXAMPLEe20c1",
          "readOnly": true,
          "resources": [
            {
              "type": "AWS::S3Express::Object",
              "ARNPrefix": "arn:aws:s3express:us-west-2:111122223333:bucket-base-name--
usw2-az1--x-s3"
            },
            {
              "accountId": "111122223333"
              "type": "AWS::S3Express::DirectoryBucket",
              "ARN": "arn:aws:s3express:us-west-2:111122223333:bucket-base-name--usw2-
az1--x-s3"
             }
           ],
           "eventType": "AwsApiCall",
           "managementEvent": false,
           "recipientAccountId": "111122223333",
           "eventCategory": "Data",
           "tlsDetails": {
             "tlsVersion": "TLSv1.3",
             "cipherSuite": "TLS_AES_128_GCM_SHA256",
             "clientProvidedHostHeader": "bucket-base-name--usw2-az1--x-s3.s3express-
usw2-az1.us-west-2.amazonaws.com"
            }
```

```
        }
```

Para utilizar las operaciones de la API de puntos de conexión zonales (operaciones de objeto o plano de datos), puede utilizar la operación de la API `CreateSession` para crear y administrar sesiones optimizadas para la autorización de solicitudes de datos con baja latencia. También se puede utilizar `CreateSession` para reducir la cantidad de registros. Para identificar qué operaciones de la API zonal se realizaron durante una sesión, puede hacer corresponder el `accessKeyId` debajo de `responseElements` en su archivo de registro `CreateSession` con el `accessKeyId` en el archivo de registro de otras operaciones de la API zonal. Para obtener más información, consulte Autorización `CreateSession`.

En el ejemplo que sigue se muestra un ejemplo de archivo de registro de CloudTrail que ilustra la operación de la API `GetObject` autenticada por `CreateSession`.

```
    {
        "eventVersion": "1.09",
        "userIdentity": {
          "type": "AssumedRole",
          "principalId": "AROAIDPPEZS35WEXAMPLE:AssumedRoleSessionName",
          "arn": "arn:aws:sts::111122223333assumed-role/RoleToBeAssumed/MySessionName",
          "accountId": "111122223333",
          "accessKeyId": "AKIAI44QH8DHBEXAMPLE",
          "sessionContext": {
            "attributes": {
              "creationDate": "2024-07-02T00:21:49Z"
            }
          }
        },
        "eventTime": "2024-07-02T00:22:01Z",
        "eventSource": "s3express.amazonaws.com",
        "eventName": "GetObject",
        "awsRegion": "us-west-2",
        "sourceIPAddress": "72.21.198.68",
        "userAgent": "aws-sdk-java/2.25.66 Linux/5.10.216-225.855.amzn2.x86_64
 OpenJDK_64-Bit_Server_VM/17.0.11+9-LTS Java/17.0.11 vendor/Amazon.com_Inc. md/internal
 exec-env/AWS_Lambda_java17 io/sync http/Apache cfg/retry-mode/legacy",
        "requestParameters": {
          "bucketName": "bucket-base-name--usw2-az1--x-s3",
          "x-amz-checksum-mode": "ENABLED",
          "Host": "bucket-base-name--usw2-az1--x-s3.s3express-usw2-az1.us-
west-2.amazonaws.com",
          "key": "test-get-obj-with-checksum"
```

```
        },
        "responseElements": null,
        "additionalEventData": {
          "SignatureVersion": "Sigv4",
          "CipherSuite": "TLS_AES_128_GCM_SHA256",
          "bytesTransferredIn": 0,
          "AuthenticationMethod": "AuthHeader",
          "x-amz-id-2": "oOy6w8K7LFsyFN",
          "bytesTransferredOut": 9,
          "availabilityZone": "usw2-az1",
          "sessionModeApplied": "ReadWrite"
         },
          "requestID": "28d2faaf-3319-4649-998d-EXAMPLE72818",
          "eventID": "694d604a-d190-4470-8dd1-EXAMPLEe20c1",
          "readOnly": true,
          "resources": [
            {
              "type": "AWS::S3Express::Object",
              "ARNPrefix": "arn:aws:s3express:us-west-2:111122223333:bucket-base-name--
usw2-az1--x-s3"
            },
            {
              "accountId": "111122223333",
              "type": "AWS::S3Express::DirectoryBucket",
              "ARN": "arn:aws:s3express:us-west-2:111122223333:bucket-base-name--usw2-
az1--x-s3"
             }
           ],
           "eventType": "AwsApiCall",
           "managementEvent": false,
           "recipientAccountId": "111122223333",
           "eventCategory": "Data",
           "tlsDetails": {
             "tlsVersion": "TLSv1.3",
             "cipherSuite": "TLS_AES_128_GCM_SHA256",
             "clientProvidedHostHeader": "bucket-base-name--usw2-az1--x-s3.s3express-
usw2-az1.us-west-2.amazonaws.com"
            }
          }
```


En el ejemplo de archivo de registro `GetObject` anterior, el `accessKeyId` (AKIAI44QH8DHBEXAMPLE) se corresponde con el `accessKeyId` que aparece debajo de

`responseElements` en el ejemplo del archivo de registro de CreateSession. El `accessKeyId` correspondiente indica la sesión en la que se realizó la operación `GetObject`.

Optimización del rendimiento de Amazon S3 Express One Zone

Amazon S3 Express One Zone es una clase de almacenamiento de S3 en una única zona de disponibilidad (AZ) de alto rendimiento que está diseñada específicamente para ofrecer acceso constante a los datos en milisegundos de un solo dígito para los datos a los que accede para las aplicaciones sensibles a la latencia. S3 Express One Zone es la primera clase de almacenamiento de S3 que le da la opción de coubicar recursos de almacenamiento de objetos de alto rendimiento e informáticos de AWS, como Amazon Elastic Compute Cloud, Amazon Elastic Kubernetes Service y Amazon Elastic Container Service, dentro de una única zona de disponibilidad. La ubicación conjunta de los recursos informáticos y de almacenamiento optimiza el rendimiento y los costes de computación y proporciona una mayor velocidad de procesamiento de datos.

S3 Express One Zone ofrece una elasticidad de rendimiento similar a la de otras clases de almacenamiento de S3, pero con latencias uniformes de solicitud de lectura y escritura de milisegundos de un solo dígito del primer byte, hasta 10 veces más rápidas que las de S3 Standard. S3 Express One Zone está diseñado desde cero para admitir un rendimiento de ráfaga hasta niveles de agregación muy elevados. La clase de almacenamiento S3 Express One Zone utiliza una arquitectura personalizada para optimizar el rendimiento y brindar una latencia de solicitud baja y uniforme mediante el almacenamiento de datos en hardware de alto rendimiento. Se ha mejorado el protocolo de objetos de S3 Express One Zone para agilizar la autenticación y la sobrecarga de metadatos.

Para aumentar aún más la velocidad de acceso y admitir cientos de miles de solicitudes por segundo, S3 Express One Zone almacena datos en un nuevo tipo de bucket: un bucket de directorio de Amazon S3. Cada bucket de directorio de S3 puede admitir cientos de miles de transacciones por segundo (TPS).

La combinación de hardware y software de alto rendimiento diseñados específicamente, que ofrece una velocidad de acceso en un dígito de milisegundos a los datos y buckets de directorio que se escalan para un gran número de transacciones por segundo, convierte a S3 Express One Zone en la mejor clase de almacenamiento de Amazon S3 para operaciones que requieren un uso intensivo de solicitudes o aplicaciones de rendimiento crítico.

En los siguientes temas se describen las directrices y patrones de diseño recomendados a fin de optimizar el rendimiento con aplicaciones que usan la clase de almacenamiento S3 Express One Zone.

Temas

Directrices de rendimiento y patrones de diseño para S3 Express One Zone

Al crear aplicaciones que cargan y recuperan objetos de Amazon S3 Express One Zone, siga las directrices de nuestras prácticas recomendadas para optimizar el rendimiento. Para usar la clase de almacenamiento S3 Express One Zone, debe crear un bucket de directorio de S3. La clase de almacenamiento S3 Express One Zone no se admite para su uso con los buckets de uso general de S3.

Para ver las directrices de rendimiento para el resto de clases de almacenamiento de Amazon S3 y los buckets de uso general de S3, consulte Prácticas recomendadas para patrones de diseño: optimizar el rendimiento de Amazon S3.

Para obtener el mejor rendimiento para su aplicación al utilizar la clase de almacenamiento S3 Express One Zone y los buckets de directorio, recomendamos las siguientes directrices y patrones de diseño.

Temas

Coubique el almacenamiento S3 Express One Zone con sus recursos informáticos de AWS

Cada bucket de directorio se almacena en una única zona de disponibilidad que seleccione al crear el bucket. Para empezar, puede crear un nuevo bucket de directorio en una zona de disponibilidad local para sus cargas de trabajo o recursos informáticos. A continuación, puede iniciar de inmediato lecturas y escrituras de muy baja latencia. Los buckets de directorio son los primeros buckets de S3 en los que puede elegir la zona de disponibilidad en una Región de AWS para reducir la latencia entre la computación y el almacenamiento.

Si accede a los buckets de directorio de todas las zonas de disponibilidad, la latencia aumentará. Para optimizar el rendimiento, le recomendamos que acceda a un bucket de directorio desde instancias de Amazon Elastic Container Service, Amazon Elastic Kubernetes Service y Amazon Elastic Compute Cloud que estén ubicadas en la misma zona de disponibilidad siempre que sea posible.

Buckets de directorio

Cada bucket de directorio puede admitir cientos de miles de transacciones por segundo (TPS). A diferencia de los buckets de uso general, los buckets de directorio organizan las claves jerárquicamente en directorios en lugar de prefijos. Un prefijo es una cadena de caracteres al principio del nombre de la clave de objeto. Puede considerar los prefijos como una forma de organizar los datos similar a los directorios. No obstante, los prefijos no son directorios.

Los prefijos organizan los datos en un espacio de nombres plano dentro de buckets de uso general y no hay límites en cuanto al número de prefijos dentro de un bucket de este tipo. Cada prefijo puede recibir al menos 3500 solicitudes `PUT/POST/DELETE` o 5500 solicitudes `GET/HEAD` por segundo. También puede paralelizar las solicitudes entre varios prefijos para escalar el rendimiento. Sin embargo, este escalado, en el caso de las operaciones de lectura y escritura, se produce gradualmente y no es instantánea. A medida que los buckets de uso general escalan según la nueva tasa de solicitudes más elevada, es posible que aparezcan algunos errores de código de estado HTTP 503 (servicio no disponible).

Con un espacio de nombres jerárquico, el delimitador de la clave del objeto es importante. El único delimitador admitido es una barra inclinada (`/`). Los directorios se determinan mediante los límites de los delimitadores. Por ejemplo, la clave del objeto `dir1/dir2/file1.txt` hace que los directorios `dir1/` y `dir2/` se creen automáticamente y que el objeto `file1.txt` se añada al directorio `/dir2` de la ruta `dir1/dir2/file1.txt`.

Los directorios que se crean cuando los objetos se cargan en los buckets de directorio no tienen límites de TPS por prefijo y se ajustan automáticamente a una escala previa para reducir la posibilidad de que se produzcan errores HTTP 503 (servicio no disponible). Este escalado automático permite a sus aplicaciones paralelizar las solicitudes de lectura y escritura dentro de los directorios y entre ellos, según sea necesario.

Uso en paralelo de solicitudes de escalado horizontal del bucket de directorio

Puede lograr el mejor rendimiento emitiendo varias solicitudes simultáneas a los buckets de directorio para distribuir las solicitudes en conexiones independientes y maximizar así el ancho de banda accesible. S3 Express One Zone no tiene límites en cuanto al número de conexiones que se realizan en su bucket de directorio. Los directorios individuales pueden escalar el rendimiento de forma horizontal y automática cuando se realizan grandes cantidades de escrituras simultáneas en el mismo directorio.

Cuando se crea inicialmente una clave de objeto y su nombre de clave incluye un directorio, el directorio se crea automáticamente para el objeto. Las cargas de objetos posteriores a ese mismo directorio no requieren que se cree el directorio, lo que reduce la latencia de las cargas de objetos a los directorios existentes.

Aunque se admiten estructuras de directorios superficiales y exhaustivas para almacenar objetos dentro de un bucket de directorio, los buckets de directorio se escalan automáticamente de forma horizontal, con una latencia más baja en cargas simultáneas en el mismo directorio o en elementos secundarios del directorio paralelo.

Uso de la autenticación basada en sesiones

S3 Express One Zone y los buckets de directorio admiten un nuevo mecanismo de autorización basado en sesiones para autenticar y autorizar las solicitudes a un bucket de directorio. Con la autenticación basada en sesiones, los SDK de AWS utilizan automáticamente la operación de la API `CreateSession` para crear un token de sesión temporal que se puede utilizar para una autorización de baja latencia de las solicitudes de datos a un bucket de directorio.

Los SDK de AWS utilizan la operación de la API `CreateSession` para solicitar credenciales temporales y, a continuación, crean y actualizan automáticamente los tokens en su nombre cada 5 minutos. Para aprovechar los beneficios de rendimiento de la clase de almacenamiento S3 Express One Zone, le recomendamos que utilice los SDK de AWS para iniciar y administrar la solicitud de la API `CreateSession`. Para obtener más información acerca de este modelo basado en sesiones, consulte Autorización de `CreateSession`.

Prácticas recomendadas adicionales para la suma de comprobación de S3

S3 Express One Zone le ofrece la opción de elegir el algoritmo de suma de comprobación que se utiliza para validar los datos durante la carga o descarga. Puede seleccionar uno de los siguientes algoritmos de comprobación de integridad de datos Secure Hash Algoritms (SHA) o Cyclic Redundancy Check (CRC): CRC32, CRC32C, SHA-1 y SHA-256. Las sumas de comprobación basadas en MD5 no son compatibles con la clase de almacenamiento S3 Express One Zone.

CRC32 es la suma de comprobación predeterminada que utilizan los SDK de AWS al transmitir datos hacia o desde S3 Express One Zone. Recomendamos utilizar CRC32 y CRC32C para obtener el mejor rendimiento con la clase de almacenamiento S3 Express One Zone.

Utilice la versión más reciente de los SDK de AWS y de las bibliotecas de tiempo de ejecución comunes

Varios de los SDK de AWS también incluyen las bibliotecas de AWS Common Runtime (CRT) para acelerar aún más el rendimiento en los clientes de S3. Estos SDK incluyen el AWS SDK for Java 2.x, el AWS SDK for C++ y el AWS SDK for Python (Boto3). El cliente S3 basado en CRT transfiere objetos desde y hacia S3 Express One Zone con una mejora del rendimiento y la fiabilidad, ya que utiliza automáticamente la operación de la API de carga multiparte y las búsquedas por rango de bytes para automatizar el escalado horizontal de conexiones.

Para lograr el máximo rendimiento con la clase de almacenamiento S3 Express One Zone, recomendamos el uso de la versión más reciente de los SDK de AWS que incluyen las bibliotecas CRT o el uso de AWS Command Line Interface (AWS CLI).

Soluciones de problemas de rendimiento

Reintento de solicitudes de aplicaciones sensibles a la latencia

S3 Express One Zone está diseñado específicamente para ofrecer niveles uniformes de alto rendimiento sin realizar ajustes adicionales. Sin embargo, establecer valores de tiempo de espera y reintentos agresivos puede ayudar aún más a lograr una latencia y un rendimiento uniformes. Los SDK de AWS cuentan con un tiempo de espera configurable y valores de reintento que puede ajustar a las tolerancias de su aplicación específica.

Emparejamiento de bibliotecas de AWS Common Runtime (CRT) y tipos de instancias de Amazon EC2

Las aplicaciones que realizan un gran número de operaciones de lectura y escritura probablemente necesitan más memoria o capacidad de computación que las aplicaciones que no. Al lanzar sus instancias de Amazon Elastic Compute Cloud (Amazon EC2) para su carga de trabajo de rendimiento exigente, seleccione los tipos de instancia que tengan la cantidad de estos recursos que necesita su aplicación. El almacenamiento de alto rendimiento S3 Express One Zone se empareja de forma ideal con tipos de instancias más grandes y más nuevos, con una mayor cantidad de memoria del sistema y CPU y GPU más potentes que pueden aprovechar el almacenamiento de mayor rendimiento. También recomendamos utilizar las versiones más recientes de los SDK de AWS habilitados para CRT, que pueden acelerar mejor las solicitudes de lectura y escritura en paralelo.

Uso de la autenticación basada en sesiones en los SDK de AWS en lugar de las API de REST de HTTP

Con Amazon S3, también es posible optimizar el rendimiento cuando se utilizan solicitudes de API de REST de HTTP siguiendo las mismas prácticas recomendadas que forman parte de los SDK de AWS. Sin embargo, con el mecanismo de autorización y autenticación basado en sesiones que utiliza S3 Express One Zone, le recomendamos fehacientemente que utilice los SDK de AWS para administrar `CreateSession` y su token de sesión administrada. Los SDK de AWS crean y actualizan automáticamente los tokens en su nombre mediante la operación de la API `CreateSession`. El uso de `CreateSession` ahorra la latencia de ida y vuelta por solicitud al AWS Identity and Access Management (IAM) para autorizar cada solicitud.

Desarrollo con S3 Express One Zone

Amazon S3 Express One Zone es la primera clase de almacenamiento de S3 en la que se puede seleccionar una única zona de disponibilidad con la opción de coubicar su almacenamiento de objetos junto con sus recursos informáticos, lo que brinda la mayor velocidad de acceso posible. Con la clase de almacenamiento S3 Express One Zone, utiliza los buckets de directorio de S3 para almacenar sus datos. Cada bucket de directorio utiliza la clase de almacenamiento S3 Express One Zone para almacenar los objetos en una única zona de disponibilidad que puede seleccionar al crear el bucket.

Una vez que haya creado su bucket de directorio, podrá comenzar de inmediato con las lecturas y escrituras de muy baja latencia. Puede comunicarse con su bucket de directorio mediante una conexión de punto de conexión a través de una nube privada virtual (VPC), o puede usar

operaciones de API zonales y regionales para administrar los objetos y los bucket de directorio. También puede usar la clase de almacenamiento S3 Express One Zone mediante la consola de Amazon S3, AWS Command Line Interface (AWS CLI), los SDK de AWS y la API de REST de Amazon S3.

La clase de almacenamiento Amazon S3 Express One Zone está diseñada para ofrecer una disponibilidad del 99,95 % dentro de una única zona de disponibilidad y está respaldada por el contrato de nivel de servicio de Amazon S3. Con S3 Express One Zone, sus datos se almacenan de forma redundante en varios dispositivos dentro de una única zona de disponibilidad. S3 Express One Zone está diseñado para controlar los fallos simultáneos de dispositivos mediante la detección y la reparación rápidas de cualquier redundancia perdida. Si el dispositivo existente detecta un fallo, S3 Express One Zone transfiere automáticamente las solicitudes a los nuevos dispositivos dentro de una zona de disponibilidad. Esta redundancia ayuda a garantizar el acceso ininterrumpido a los datos dentro de una zona de disponibilidad.

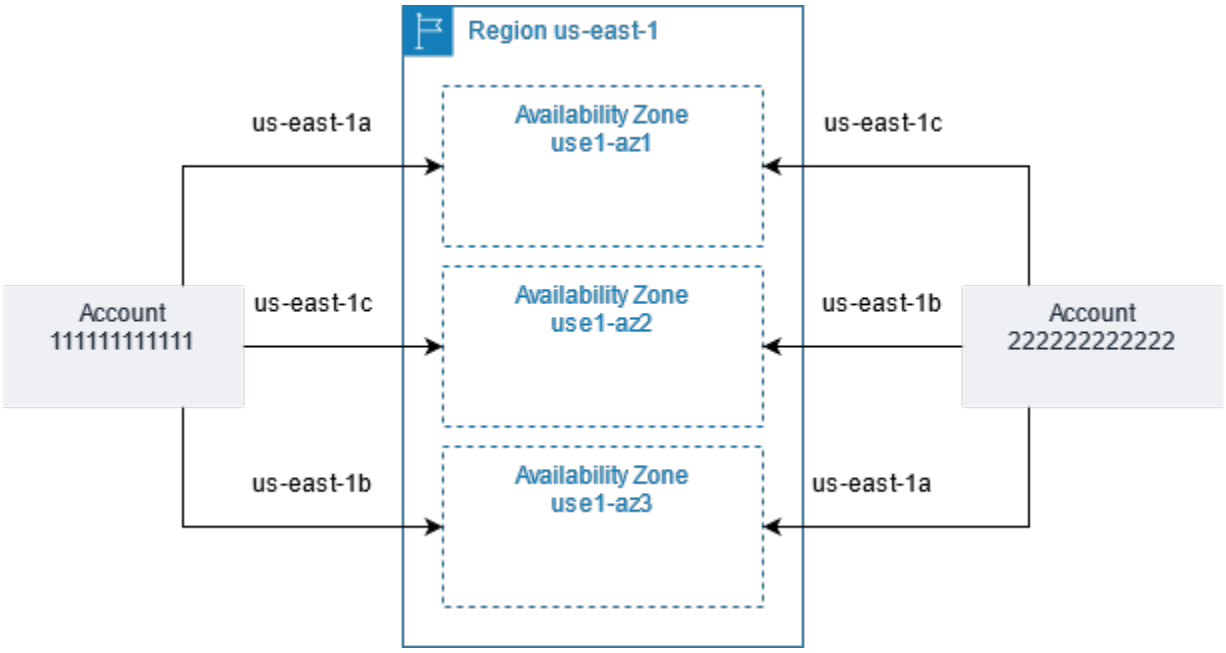
Temas

Zonas y regiones de disponibilidad de S3 Express One Zone

Una zona de disponibilidad consiste en uno o varios centros de datos discretos con alimentación, redes y conectividad redundantes en una Región de AWS. Para optimizar las recuperaciones de baja latencia, los objetos de la clase de almacenamiento Amazon S3 Express One Zone se almacenan de forma redundante en buckets de directorio de S3 en una única zona de disponibilidad local para su carga de trabajo informática. Al crear un bucket de directorio, debe elegir la zona de disponibilidad y la Región de AWS donde se colocará el bucket.

AWS asigna las zonas de disponibilidad física aleatoriamente a los nombres de las zonas de disponibilidad de cada Cuenta de AWS. Este enfoque facilita la distribución de los recursos entre las zonas de disponibilidad de una Región de AWS, para reducir la probabilidad de que los recursos se concentren en la primera zona de disponibilidad de cada región. Como resultado, es posible que la zona de disponibilidad `us-east-1a` de su Cuenta de AWS no se refiera a la misma ubicación física que `us-east-1a` de otra Cuenta de AWS. Para obtener más información, consulte Regiones y zonas de disponibilidad en la Guía del usuario de Amazon EC2.

Para coordinar las zonas de disponibilidad entre cuentas, debe usar el ID de AZ, que es un identificador único y constante de una zona de disponibilidad. Por ejemplo, `use1-az1` es un ID de zona de disponibilidad de la región `us-east-1` y tiene la misma ubicación física en cada Cuenta de AWS. En la siguiente ilustración se muestra cómo los ID de zona de disponibilidad son los mismos para todas las cuentas, a pesar de los nombres de las zonas de disponibilidad pueden asignarse de forma diferente para cada cuenta.



Con S3 Express One Zone, sus datos se almacenan de forma redundante en varios dispositivos dentro de una única zona de disponibilidad. S3 Express One Zone está diseñado para ofrecer una disponibilidad del 99,95 % dentro de una única zona de disponibilidad y está respaldado por el contrato de nivel de servicio de Amazon S3. Para obtener más información, consulte Una sola zona de disponibilidad

S3 Express One Zone se admite en las siguientes regiones y zonas de disponibilidad:

Regiones y zonas de disponibilidad compatibles con S3 Express One Zone

Nombre de la región	Código de región	ID de zona de disponibilidad
Este de EE. UU. (Norte de Virginia)	`us-east-1`	`use1-az4`
		`use1-az5`

Nombre de la región	Código de región	ID de zona de disponibilidad
		use1-az6
Oeste de EE. UU. (Oregón)	us-west-2	usw2-az1
		usw2-az3
		usw2-az4
Asia-Pacífico (Tokio)	ap-northeast-1	apne1-az1
		apne1-az4
Europa (Estocolmo)	eu-north-1	eun1-az1
		eun1-az2
		eun1-az3

Puntos de conexión regionales y zonales

Para acceder a los puntos de conexión regionales y zonales de Amazon S3 Express One Zone desde su nube privada virtual (VPC), puede usar puntos de conexión de VPC de puerta de enlace. Después de crear un punto de conexión de la puerta de enlace, puede agregarlo como destino en la tabla de enrutamiento para el tráfico destinado desde la VPC a S3 Express One Zone. El uso de puntos de enlace de gateway no supone ningún cargo adicional. Para obtener más información acerca de cómo configurar los puntos de conexión de VPC de puerta de enlace, consulte Redes para S3 Express One Zone.

Cuando trabaja con S3 Express One Zone, las operaciones de la API a nivel de bucket (plano de control) están disponibles a través de un punto de enlace regional y se denominan operaciones de API de punto final regional. Algunos ejemplos de operaciones de la API de puntos de conexión regionales son `CreateBucket` y `DeleteBucket`.

Tras crear un bucket de directorio, puede utilizar las operaciones de la API de puntos de conexión zonales (de nivel de objeto o plano de datos) para cargar y administrar los objetos en su bucket de

directorio. Las operaciones de la API de puntos de conexión zonales están disponibles a través de un punto de conexión zonal. Algunos ejemplos de operaciones de API zonales son `PutObject` y `CopyObject`.

Trabajo con S3 Express One Zone mediante la consola de S3, la AWS CLI y los AWS SDK

Puede trabajar con la clase de almacenamiento S3 Express One Zone y los buckets de directorio mediante los SDK de AWS, la consola de Amazon S3, AWS Command Line Interface (AWS CLI) y la API de REST de Amazon S3.

Consola de S3

Para empezar a usar la consola de S3, siga estos pasos:

- Crear un bucket de directorio
- Vaciado de un bucket de directorio
- Eliminar un bucket de directorio

Para ver un tutorial completo, consulte Tutorial: introducción a S3 Express One Zone.

SDK de AWS

S3 Express One Zone es compatible con los siguientes SDK de AWS:

- 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
- AWS SDK para Kotlin
- AWS SDK para Rust

Cuando trabaje con S3 Express One Zone, le recomendamos que utilice la versión más reciente de los SDK de AWS. Los SDK de AWS compatibles con S3 Express One Zone controlan el establecimiento, la actualización y la finalización de la sesión en su nombre. Esto significa que puede empezar a usar las operaciones de la API inmediatamente después de descargar e instalar los SDK de AWS y configurar los permisos de IAM necesarios. Para obtener más información, consulte AWS Identity and Access Management (IAM) para S3 Express One Zone.

Para obtener información sobre los SDK de AWS (por ejemplo: cómo descargarlos e instalarlos), consulte Herramientas para crear en AWS.

Para ver ejemplos del SDK de AWS, consulte lo siguiente:

- Crear un bucket de directorio
- Vaciado de un bucket de directorio
- Eliminar un bucket de directorio

AWS Command Line Interface (AWS CLI)

Puede usar la AWS Command Line Interface (AWS CLI) para crear buckets de directorio y utilizar las operaciones de la API de puntos de conexión regionales y zonales compatibles con S3 Express One Zone.

Para empezar con la AWS CLI, consulte Introducción a la AWS CLI en la Referencia de comandos de AWS CLI.

> **Note**
>
> Para usar buckets de directorio con los comandos `aws s3` de alto nivel, actualice su AWS CLI a la versión más reciente. Para obtener más información sobre cómo instalar y configurar la AWS CLI, consulte Instalar o actualizar la última versión de la AWS CLI en la Referencia de comandos de AWS CLI.

Para ver ejemplos de AWS CLI, consulte lo siguiente:

- Crear un bucket de directorio
- Vaciado de un bucket de directorio
- Eliminar un bucket de directorio

Operaciones de la API de S3 Express One Zone

La clase de almacenamiento Amazon S3 Express One Zone admite operaciones de la API de puntos de conexión regionales (nivel de bucket o plano de control) y zonales (nivel de objeto o plano de datos). Para obtener más información, consulte Redes para S3 Express One Zone y Puntos de conexión y puntos de conexión de VPC de puerta de enlace.

Operaciones de la API de puntos de conexión regionales

S3 Express One Zone admite las siguientes operaciones de la API de puntos de conexión regionales:

- CreateBucket
- DeleteBucket
- DeleteBucketPolicy
- GetBucketPolicy
- ListDirectoryBuckets
- PutBucketPolicy

Operaciones de la API de puntos de conexión zonales

S3 Express One Zone admite las siguientes operaciones de la API de punto de conexión zonales:

- CreateSession
- CopyObject
- DeleteObject
- DeleteObjects
- GetObject
- GetObjectAttributes
- HeadBucket
- HeadObject
- ListObjectsV2
- PutObject
- AbortMultipartUpload
- CompleteMultiPartUpload
- CreateMultipartUpload

- ListMultipartUploads
- ListParts
- UploadPart
- UploadPartCopy

Administración del acceso a datos con puntos de acceso de Amazon S3

Los puntos de acceso de Amazon S3 facilitan el acceso a datos para cualquier servicio de AWS o aplicación de cliente que almacena datos en S3. Los puntos de acceso son puntos de enlace de red con nombre y asociados a los buckets que se pueden utilizar para realizar operaciones con objetos de S3, como `GetObject` y `PutObject`. Cada punto de acceso tiene permisos y controles de red distintos que S3 aplica a cualquier solicitud que se realice a través de ese punto de acceso. Cada punto de acceso aplica una política de punto de acceso personalizada que funciona en conjunción con la política de bucket asociada al bucket subyacente. Puede configurar cualquier punto de acceso para que acepte solo las solicitudes procedentes de una nube privada virtual (VPC) con el fin de restringir el acceso a los datos de Amazon S3 a una red privada. También puede configurar los parámetros de bloqueo de acceso público para cada punto de acceso.

> **Note**
>
> - Solo se pueden utilizar puntos de acceso para realizar operaciones con objetos. No se pueden utilizar puntos de acceso para realizar otras operaciones de Amazon S3, como modificar o eliminar buckets. Para obtener una lista completa de las operaciones de S3 que admiten puntos de acceso, consulte Compatibilidad del punto de acceso con servicios de AWS.
> - Los puntos de acceso funcionan con algunos servicios y características de AWS, pero no con todos. Por ejemplo, no puede configurar la replicación entre regiones para que se lleve a cabo a través de un punto de acceso. Para obtener una lista completa de servicios de AWS compatibles con los puntos de acceso de S3, consulte Compatibilidad del punto de acceso con servicios de AWS.

En esta sección se explica cómo trabajar con puntos de acceso de Amazon S3. Para obtener información sobre cómo trabajar con buckets, consulte Descripción general de los buckets. Para obtener información acerca del uso de objetos, consulte Información general de los objetos de Amazon S3.

Temas

- Crear puntos de acceso
- Usar puntos de acceso
- Restricciones y limitaciones de los puntos de acceso

Configurar las políticas de IAM para el uso de puntos de acceso

Los puntos de acceso de Amazon S3 admiten políticas de recursos de AWS Identity and Access Management (IAM) que permiten controlar el uso del punto de acceso en función del recurso, del usuario o de otras condiciones. Para que una aplicación o un usuario puedan acceder a objetos a través de un punto de acceso, tanto el punto de acceso como el bucket subyacente deben permitir la solicitud.

> **Important**
>
> Agregar un punto de acceso de S3 a un bucket no cambia el comportamiento del bucket cuando se accede a él directamente mediante el nombre del bucket o el nombre de recurso de Amazon (ARN). Todas las operaciones existentes respecto al bucket continuarán funcionando como antes. Las restricciones que se incluyen en una política de punto de acceso solo se aplican a las solicitudes realizadas a través de ese punto de acceso.

Cuando utilice políticas de recurso de IAM, asegúrese de resolver advertencias de seguridad, errores, advertencias generales y sugerencias de AWS Identity and Access Management Access Analyzer antes de guardar la política. IAM Access Analyzer ejecuta verificaciones de política para validarla contra la Gramática de la política de IAM y las prácticas recomendadas. Estas verificaciones generan hallazgos y proporcionan recomendaciones para ayudarle a crear políticas funcionales y que se ajustan a las prácticas recomendadas de seguridad.

Para obtener más información sobre la validación de políticas utilizando IAM Access Analyzer, consulte Validación de políticas de IAM Access Analyzer en la Guía del usuario de IAM. Para ver una lista de advertencias, errores y sugerencias que devuelve IAM Access Analyzer, consulte Referencia de verificación de políticas de IAM Access Analyzer.

Ejemplos de políticas de puntos de acceso

En los ejemplos siguientes se muestra cómo crear políticas de IAM para controlar las solicitudes realizadas a través de un punto de acceso.

> **Note**
>
> Los permisos concedidos en una política de puntos de acceso se aplican solo si el bucket subyacente también permite el mismo acceso. Puede lograr esto de dos maneras:
>
> 1. (Recomendado) Delegue el control de acceso del bucket al punto de acceso como se describe en Delegar el control de acceso a los puntos de acceso.
> 2. Agregue los mismos permisos contenidos en la política de puntos de acceso a la política del bucket subyacente. En el ejemplo 1 de política de puntos de acceso se muestra cómo modificar la política de bucket subyacente para permitir el acceso necesario.

Example 1: Conceder política de punto de acceso

La siguiente política de punto de acceso concede al usuario de IAM *Jane* de la cuenta *123456789012* permisos para objetos GET y PUT con el prefijo *Jane/* a través del punto de acceso *my-access-point* de la cuenta *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**
>
> Para que la política de punto de acceso conceda efectivamente acceso a *Jane*, el bucket subyacente también debe permitir el mismo acceso a *Jane*. Puede delegar el control de acceso desde el bucket al punto de acceso como se describe en Delegar el control de acceso a los puntos de acceso. O puede agregar la siguiente política al bucket subyacente

para conceder los permisos necesarios a Jane. Tenga en cuenta que la entrada `Resource` difiere entre las políticas de punto de acceso y de bucket.

```
{
    "Version": "2012-10-17",
    "Statement": [
    {
        "Effect": "Allow",
        "Principal": {
            "AWS": "arn:aws:iam::123456789012:user/Jane"
        },
        "Action": ["s3:GetObject", "s3:PutObject"],
        "Resource": "arn:aws:s3:::amzn-s3-demo-bucket1/Jane/*"
    }]
}
```

Example 2: Política de punto de acceso con condición de etiqueta

La siguiente política de punto de acceso concede al usuario de IAM *Mateo* de la cuenta *123456789012* permisos para objetos GET a través del punto de acceso *my-access-point* de la cuenta *123456789012* que tengan la clave de etiqueta *data* establecida con un valor de *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: Política de puntos de acceso que permite el listado de buckets

La siguiente política de punto de acceso concede al usuario de IAM `Arnav` de la cuenta *123456789012* permiso para consultar los objetos contenidos en el punto de acceso subyacente del bucket *my-access-point* de la cuenta *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: Política de control de servicios

La política de control de servicio siguiente requiere que todos los puntos de acceso nuevos se creen con el origen de red de nube privada virtual (VPC). Con esta política en vigor, los usuarios de la organización no pueden crear nuevos puntos de acceso a los que se pueda acceder desde Internet.

```
{
    "Version": "2012-10-17",
    "Statement": [
    {
        "Effect": "Deny",
        "Action": "s3:CreateAccessPoint",
        "Resource": "*",
        "Condition": {
            "StringNotEquals": {
                "s3:AccessPointNetworkOrigin": "VPC"
            }
        }
    }]
}
```

Example 5: Política de bucket para limitar las operaciones de S3 a los orígenes de red VPC

La siguiente política de bucket limita el acceso a todas las operaciones de objetos de S3 para el bucket *amzn-s3-demo-bucket* a los puntos de acceso cuyo origen de red sea VPC.

> **Important**
>
> Antes de utilizar una instrucción como la que se muestra en este ejemplo, asegúrese de que no necesita usar características que no sean compatibles con los puntos de acceso, como la replicación entre regiones.

```
{
    "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:::amzn-s3-demo-bucket/*",
            "Condition": {
                "StringNotEquals": {
                    "s3:AccessPointNetworkOrigin": "VPC"
                }
            }
        }
    ]
}
```

Claves de condición

Los puntos de acceso de S3 tienen claves de condición que se pueden utilizar en las políticas de IAM para controlar el acceso a los recursos. Las siguientes claves de condición representan solo una parte de una política de IAM. Para ver ejemplos de política completos, consulte Ejemplos de políticas de puntos de acceso, the section called “Delegar el control de acceso a los puntos de acceso” y the section called “Concesión de permisos para puntos de acceso entre cuentas”.

`s3:DataAccessPointArn`

Este ejemplo muestra una cadena que puede usar para obtener coincidencias en el ARN de un punto de acceso. En el siguiente ejemplo, se encuentran todas las concordancias con puntos de acceso de la Cuenta de AWS *123456789012* de la región *us-west-2*:

```
"Condition" : {
    "StringLike": {
        "s3:DataAccessPointArn": "arn:aws:s3:us-west-2:123456789012:accesspoint/*"
    }
}
```

`s3:DataAccessPointAccount`

Este ejemplo muestra un operador de cadena que puede utilizar para hallar coincidencias del ID de cuenta del propietario de un punto de acceso. En el siguiente ejemplo, se encuentran todas las concordancias de puntos de acceso que son propiedad de la Cuenta de AWS *123456789012*.

```
"Condition" : {
    "StringEquals": {
        "s3:DataAccessPointAccount": "123456789012"
    }
```

```
}
```

`s3:AccessPointNetworkOrigin`

En este ejemplo, se muestra un operador de cadena que puede utilizar para hallar coincidencias del origen de red, ya sea `Internet` o VPC. En el ejemplo siguiente, solo se hallarán las coincidencias de los puntos de acceso cuyo origen sea VPC.

```
"Condition" : {
    "StringEquals": {
        "s3:AccessPointNetworkOrigin": "VPC"
    }
}
```

Para obtener más información sobre el uso de claves de condición con Amazon S3, consulte Actions, resources, and condition keys for Amazon S3 en la Referencia de autorización de servicios.

Delegar el control de acceso a los puntos de acceso

Puede delegar el control de acceso de un bucket en los puntos de acceso del bucket. La política de bucket de ejemplo siguiente permite el acceso completo a todos los puntos de acceso de la cuenta del propietario del bucket. Por lo tanto, todo el acceso a este bucket está controlado por las políticas asociadas a sus puntos de acceso. Recomendamos configurar los buckets de esta manera para todos los casos de uso que no requieran acceso directo al bucket.

Example 6: Política de bucket que delega el control de acceso a los puntos de acceso

```
{
    "Version": "2012-10-17",
    "Statement" : [
    {
        "Effect": "Allow",
        "Principal" : { "AWS": "*" },
        "Action" : "*",
        "Resource" : [ "Bucket ARN", "Bucket ARN/*"],
        "Condition": {
            "StringEquals" : { "s3:DataAccessPointAccount" : "Bucket owner's account
 ID" }
        }
    }]
```

```
}
```

Concesión de permisos para puntos de acceso entre cuentas

Para crear un punto de acceso a un bucket que es propiedad de otra cuenta, primero debe crear el punto de acceso especificando el nombre del bucket y el ID del propietario de la cuenta. A continuación, el propietario del bucket debe actualizar la política del bucket para autorizar las solicitudes desde el punto de acceso. La creación de un punto de acceso es similar a crear un CNAME de DNS, ya que el punto de acceso no proporciona acceso al contenido del bucket. Todo el acceso al bucket está controlado por la política de bucket. La siguiente política de bucket de ejemplo permite solicitudes `GET` y `LIST` en el bucket desde un punto de acceso que es propiedad de una Cuenta de AWS de confianza.

Reemplace *`ARN del bucket`* por el ARN del bucket.

Example 7: Política de bucket que delega permisos en otra Cuenta de 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" }
        }
    }]
}
```

Crear puntos de acceso

Amazon S3 proporciona funcionalidad para crear y administrar puntos de acceso. Puede crear puntos de acceso de S3 mediante la AWS Management Console, AWS Command Line Interface (AWS CLI), los SDK de AWS o la API de REST de Amazon S3.

De forma predeterminada, puede crear hasta 10 000 puntos de acceso por región para cada una de sus Cuentas de AWS. Si necesita más de 10 000 puntos de acceso para una sola cuenta en una

sola región, puede solicitar un aumento de la cuota de servicio. Para obtener más información sobre las cuotas de servicio y solicitar un aumento, consulte Cuotas de servicio de AWS en la Referencia general de AWS.

> **Note**
>
> Dado que es posible que desee publicar el nombre del punto de acceso para que otros usuarios puedan utilizarlo, evite incluir información confidencial. Los nombres de los puntos de acceso se publican en una base de datos de acceso público conocida como sistema de nombres de dominio (DNS).

Reglas para asignar nombres a los puntos de acceso de Amazon S3

Los nombres de los puntos de acceso deben cumplir las condiciones siguientes:

- Debe ser único para cada región y Cuenta de AWS.
- Debe cumplir las restricciones de nomenclatura de DNS.
- Debe comenzar con un número o una letra minúscula.
- Deben tener entre 3 y 50 caracteres de longitud.
- No puede comenzar ni terminar con un guion (`-`)
- No puede contener guiones bajos (`_`), letras mayúsculas ni puntos (`.`)
- No puede terminar con el sufijo `-s3alias`. Este sufijo está reservado para nombres de alias de punto de acceso. Para obtener más información, consulte Uso de un alias de estilo de bucket para su punto de acceso de bucket de S3.

Para crear un punto de acceso, consulte los temas siguientes.

Temas

- Creación de un punto de acceso
- Crear puntos de acceso restringidos a una nube privada virtual
- Administrar el acceso público a los puntos de acceso

Creación de un punto de acceso

Un punto de acceso está asociado con exactamente un bucket de Amazon S3. Si desea utilizar un bucket en la Cuenta de AWS, primero debe crear un bucket. Para obtener más información acerca de cómo se crean los buckets, consulte Creación, configuración y trabajo con buckets de Amazon S3.

También puede crear un punto de acceso entre cuentas que esté asociado a un bucket en otra Cuenta de AWS, siempre y cuando conozca el nombre del bucket y el ID de cuenta del propietario del bucket. Sin embargo, la creación de puntos de acceso entre cuentas no le permite acceder a los datos del bucket hasta que el propietario del bucket le conceda los permisos. El propietario del bucket debe conceder acceso al bucket a la cuenta del propietario del punto de acceso (su cuenta) mediante la política de bucket. Para obtener más información, consulte Concesión de permisos para puntos de acceso entre cuentas.

De forma predeterminada, puede crear hasta 10 000 puntos de acceso por región para cada una de sus Cuentas de AWS. Si necesita más de 10 000 puntos de acceso para una sola cuenta en una sola región, puede solicitar un aumento de la cuota de servicio. Para obtener más información sobre las cuotas de servicio y solicitar un aumento, consulte Cuotas de servicio de AWS en la Referencia general de AWS.

En los siguientes ejemplos, se muestra cómo crear un punto de acceso con la AWS CLI y la consola de S3. Para obtener más información acerca de cómo crear puntos de acceso mediante la API de REST, consulte CreateAccessPoint en la Referencia de la API de Amazon Simple Storage Service.

Uso de la consola de S3

Para crear un punto de acceso

1. Inicie sesión AWS Management Console Management Console y abra la consola de Amazon S3 en https://console.aws.amazon.com/s3/.
2. En la barra de navegación de la parte superior de la página, elija el nombre de la Región de AWS que aparece. A continuación, elija la región en la que desea crear un punto de acceso.
3. En el panel de navegación, elija Puntos de acceso.
4. En la página de Puntos de acceso, elija Crear punto de acceso.
5. Introduzca el nombre que desee para el punto de acceso en el campo Nombre del punto de acceso. Para obtener más información sobre cómo asignar nombres a los puntos de acceso, consulte Reglas para asignar nombres a los puntos de acceso de Amazon S3.

6. En Nombre del bucket, especifique el bucket de S3 que desea usar con el punto de acceso.

   Para usar un bucket en la cuenta, seleccione Elegir un bucket de esta cuenta y escriba o examine los nombres del bucket.

   Para usar un bucket de otra Cuenta de AWS, elija Especificar un bucket de otra cuenta e introduzca el ID Cuenta de AWS y el nombre del bucket.

   > **Note**
   >
   > Si está utilizando un bucket de otra Cuenta de AWS, el propietario del bucket debe actualizar la política del bucket para autorizar las solicitudes del punto de acceso. Para ver una política de bucket de ejemplo, consulte Concesión de permisos para puntos de acceso entre cuentas.

7. Elija un origen de red. Si elige Virtual private cloud (VPC) (Nube privada virtual [VPC]), escriba el identificador VPC ID (ID de VPC) que desea usar con el punto de acceso.

   Para obtener más información acerca de los orígenes de red para los puntos de acceso, consulte Crear puntos de acceso restringidos a una nube privada virtual.

8. En Block Public Access settings for this Access Point (Configuración de bloqueo del acceso público a este punto de acceso), seleccione la configuración de bloqueo de acceso público que desee aplicar al punto de acceso. Todas las configuraciones de bloqueo de acceso público están habilitadas de forma predeterminada para los puntos de acceso nuevos. Le recomendamos que deje todas las configuraciones habilitadas a menos que sepa que tiene una necesidad específica de desactivar cualquiera de ellas.

   > **Note**
   >
   > Después de crear un punto de acceso, no puede cambiar su configuración de bloqueo de acceso público.

   Para obtener más información sobre el uso de Amazon S3 Block Public Access con puntos de acceso, consulte Administrar el acceso público a los puntos de acceso.

9. (Opcional) En Access point policy - optional (Política de punto de acceso: opcional), especifique la política de punto de acceso. Antes de guardar la política, asegúrese de resolver las advertencias de seguridad, los errores, las advertencias generales y las sugerencias. Para

obtener más información acerca de cómo especificar una política de punto de acceso, consulte Ejemplos de políticas de puntos de acceso.

10. Elija Create access point (Crear punto de acceso).

Uso de la AWS CLI

El comando de ejemplo siguiente crea un punto de acceso denominado *example-ap* para el bucket *amzn-s3-demo-bucket* de la cuenta *111122223333*. Para crear el punto de acceso, debe enviar una solicitud a Amazon S3 que especifique lo siguiente:

- El nombre del punto de acceso. Para obtener información sobre las reglas de nomenclatura, consulte the section called “Reglas para asignar nombres a los puntos de acceso de Amazon S3”.
- El nombre del bucket al que desea asociar el punto de acceso.
- El ID de la cuenta para la Cuenta de AWS propietaria del bucket.

```
aws s3control create-access-point --name example-ap --account-id 111122223333 --
bucket amzn-s3-demo-bucket
```

Cuando cree un punto de acceso mediante un bucket en una Cuenta de AWS diferente, incluya el parámetro `--bucket-account-id`. El comando de ejemplo siguiente crea un punto de acceso en la Cuenta de AWS *111122223333*, con el bucket *amzn-s3-demo-bucket2*, que está en la Cuenta de AWS *444455556666*.

```
aws s3control create-access-point --name example-ap --account-id 111122223333 --
bucket amzn-s3-demo-bucket --bucket-account-id 444455556666
```

Crear puntos de acceso restringidos a una nube privada virtual

Al crear un punto de acceso, puede optar por hacer que sea accesible desde Internet o especificar que todas las solicitudes realizadas a través de ese punto de acceso deban originarse en una nube privada virtual (VPC) específica. Se dice que el origen de red de un punto de acceso que es accesible desde Internet es `Internet`. Se puede usar desde cualquier lugar de Internet, con sujeción a las restricciones de acceso en vigor para el punto de acceso, el bucket subyacente y los recursos relacionados, como los objetos solicitados. El origen de un punto de acceso al que solo se puede acceder desde una VPC es `VPC`. En este caso, Amazon S3 rechazará todas las solicitudes realizadas al punto de acceso cuando el origen de estas no sea esa misma VPC.

> **Important**
>
> Solo se puede especificar el origen de red de un punto de acceso en el momento de crearlo. Una vez creado el punto de acceso, ya no puede cambiar su origen de red.

Para restringir un punto de acceso de modo que sea solo VPC, incluya el parámetro `VpcConfiguration` con la solicitud de creación del punto de acceso. En el parámetro `VpcConfiguration`, especifique el ID de VPC que desee que el punto de acceso pueda utilizar. Si se realiza una solicitud a través del punto de acceso, la solicitud debe originarse en la VPC; de lo contrario, Amazon S3 la rechazará.

Puede recuperar el origen de red de un punto de acceso mediante la AWS CLI, los AWS SDK o las API de REST. Si para un punto de acceso se ha especificado una configuración de VPC, su origen de red es `VPC`. De lo contrario, el origen de red del punto de acceso es `Internet`.

Example

Ejemplo: Crear un punto de acceso que tiene restringido al acceso de VPC

En el ejemplo siguiente se crea un punto de acceso denominado `example-vpc-ap` para el bucket `example-bucket` en la cuenta `123456789012`, que permite el acceso únicamente desde la VPC `vpc-1a2b3c`. A continuación, el ejemplo comprueba que el origen de red del nuevo punto de acceso sea `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"
}
```

Para utilizar un punto de acceso con una VPC, debe modificar la política de acceso para el punto de conexión de la VPC. Los puntos de enlace de la VPC permiten que el tráfico fluya desde su VPC hasta Amazon S3. Tienen políticas de control de acceso que controlan cómo pueden interactuar con Amazon S3 los recursos dentro de la VPC. Las solicitudes de la VPC a Amazon S3 solo se realizan correctamente a través de un punto de conexión de VPC si la política de punto de conexión de VPC concede acceso tanto al punto de conexión como al bucket subyacente.

> **Note**
>
> Para que los recursos estén accesibles solo dentro de una VPC, asegúrese de crear una zona alojada privada para el punto de conexión de VPC. Para usar una zona alojada privada, modifique la configuración de VPC, de modo que los atributos de red de VPC `enableDnsHostnames` y `enableDnsSupport` estén configurados para `true`.

En el siguiente ejemplo de una instrucción de política, se configura un punto de conexión de la VPC para permitir llamadas a `GetObject` para un bucket denominado `awsexamplebucket1` y un punto de acceso denominado `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 declaración `"Resource"` de este ejemplo utiliza un nombre de recurso de Amazon (ARN) para especificar el punto de acceso. Para obtener más información acerca de los ARN de puntos de acceso, consulte Usar puntos de acceso.

Para obtener más información acerca de las políticas de punto de conexión de VPC, consulte Utilización de políticas de punto de conexión para Amazon S3 en la Guía del usuario de VPC.

Administrar el acceso público a los puntos de acceso

Los puntos de acceso de Amazon S3 admiten configuraciones de Bloqueo de acceso público independientes para cada punto de acceso. Al crear un punto de acceso, puede especificar la configuración del bloqueo de acceso público aplicable a ese punto de acceso. Para cualquier solicitud realizada a través de un punto de acceso, Amazon S3 evalúa la configuración de bloqueo de acceso público para ese punto de acceso, el bucket subyacente y la cuenta propietaria del bucket. Si alguna de estas configuraciones indica que la solicitud debe ser bloqueada, Amazon S3 rechaza la solicitud.

Para obtener más información acerca de la característica S3 Block Public Access, consulte Bloquear el acceso público a su almacenamiento de Amazon S3.

> **Important**
>
> - Todas las configuraciones de bloqueo de acceso público están habilitadas de forma predeterminada para los puntos de acceso. Debe deshabilitar explícitamente cualquier configuración que no desee aplicar a un punto de acceso.
> - Amazon S3 actualmente no admite cambiar la configuración de bloqueo de acceso público de un punto de acceso después de que se haya creado el punto de acceso.

Example

Ejemplo: Crear un punto de acceso con una configuración de bloqueo de acceso público personalizada

En este ejemplo se crea un punto de acceso denominado `example-ap` para el bucket `example-bucket` en la cuenta `123456789012` con una configuración de bloqueo de acceso público no predeterminada. A continuación, en el ejemplo se recupera la configuración del nuevo punto de acceso para comprobar su configuración de bloqueo de acceso público.

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"
}
```

Usar puntos de acceso

Puede acceder a los objetos en un bucket de Amazon S3 con un punto de acceso mediante la AWS Management Console, la AWS CLI, los SDK de AWSo las API de REST de S3.

Los puntos de acceso tienen nombres de recurso de Amazon (ARN). Los ARN de los puntos de acceso son similares a los ARN de los buckets, pero tienen tipos explícitos y llevan codificada la región del punto de acceso y el ID de la Cuenta de AWS del propietario del punto de acceso. Para obtener más información acerca de los ARN, consulte Nombres de recurso de Amazon (ARN) en la Referencia general de AWS.

Los ARN de los puntos de acceso utilizan el formato `arn:aws:s3:`*`region`*`:`*`account-id`*`:accesspoint/`*`resource`*. Por ejemplo:

- arn:aws:s3:us-west-2:123456789012:accesspoint/test representa el punto de acceso denominado `test`, propiedad de la cuenta `123456789012` de la región `us-west-2`.
- arn:aws:s3:us-west-2:123456789012:accesspoint/* representa todos los puntos de acceso de la cuenta `123456789012` de la región `us-west-2`.

Los ARN para los objetos a los que se accede a través de un punto de acceso utilizan el formato `arn:aws:s3:`*`region`*`:`*`account-id`*`:accesspoint/`*`access-point-name`*`/object/`*`resource`*. Por ejemplo:

- arn:aws:s3:us-west-2:123456789012:accesspoint/test/object/unit-01 representa el objeto `unit-01`, al que se accede a través del punto de acceso denominado `test`, que es propiedad de la cuenta `123456789012` de la región `us-west-2`.
- arn:aws:s3:us-west-2:123456789012:accesspoint/test/object/* representa todos los objetos del punto de acceso `test`, de la cuenta `123456789012` de la región `us-west-2`.
- arn:aws:s3:us-west-2:123456789012:accesspoint/test/object/unit-01/finance/* representa todos los objetos con el prefijo `unit-01/finance/` para el punto de acceso `test`, de la cuenta `123456789012` de la región `us-west-2`.

Acceso a un bucket a través de los puntos de acceso de S3

Los puntos de acceso de S3 solo admiten el direccionamiento de tipo host virtual. Para direccionar un bucket a través de un punto de acceso, utilice el siguiente formato.

```
https://AccessPointName-AccountId.s3-accesspoint.region.amazonaws.com
```

> **Note**
>
> - Si el nombre del punto de acceso incluye caracteres de guion (-), incluya los guiones en la URL e inserte otro guion antes del ID de cuenta. Por ejemplo, para utilizar un punto de acceso denominado `finance-docs` propiedad de la cuenta `123456789012` en la región `us-west-2`, la dirección URL apropiada sería `https://finance-docs-123456789012.s3-accesspoint.us-west-2.amazonaws.com`.
> - Los puntos de acceso de S3 no admiten el acceso por HTTP, solo el acceso seguro por HTTPS.

Temas

Si tiene una nube privada virtual (VPC), consulte Administración del acceso a Amazon S3 con puntos de conexión de VPC y puntos de acceso de S3.

Monitorización y registro de puntos de acceso

Amazon S3 registra las solicitudes realizadas a través de los puntos de acceso y las solicitudes realizadas a las API que administran puntos de acceso, tales como `CreateAccessPoint` y `GetAccessPointPolicy`. Para monitorear y gestionar patrones de uso, también puede configurar las métricas de solicitud de Amazon CloudWatch Logs para los puntos de acceso.

Temas

Métricas de solicitudes de CloudWatch

Para comprender y mejorar el rendimiento de las aplicaciones que utilizan puntos de acceso, puede utilizar las métricas de solicitudes de CloudWatch para Amazon S3. Las métricas de solicitudes le permiten monitorear las solicitudes de Amazon S3 para identificar rápidamente los problemas operativos y actuar en consecuencia.

De forma predeterminada, estas métricas están disponibles en el nivel del bucket. Sin embargo, puede definir un filtro para las métricas de solicitudes mediante un prefijo compartido, etiquetas de objeto o un punto de acceso. Al crear un filtro de punto de acceso, la configuración de métricas de solicitudes incluye las solicitudes al punto de acceso que especifique. Puede recibir métricas, establecer alarmas y tener acceso a paneles de para ver las operaciones en tiempo real realizadas a través de este punto de acceso.

Debe incluir métricas de solicitudes configurándolas en la consola o con la API de Amazon S3. Las métricas de solicitudes están disponibles en intervalos de 1 minuto después de un breve periodo de

latencia para procesarlas. Las métricas de solicitudes se facturan al mismo precio que las métricas personalizadas de CloudWatch. Para más información, consulte Precios de Amazon CloudWatch.

Para crear una configuración de métricas de solicitudes que filtra por punto de acceso, consulte Creación de una configuración de métricas que filtra por prefijo, etiqueta de objeto o punto de acceso.

Registros de solicitudes

Puede registrar las solicitudes realizadas a través de los puntos de acceso y las solicitudes realizadas a las API que administran puntos de acceso, tales como `CreateAccessPoint` y `GetAccessPointPolicy`, usando el registro de acceso del servidor y AWS CloudTrail.

Las entradas de registro de CloudTrail para solicitudes realizadas a través de puntos de acceso incluirán el ARN del punto de acceso en la sección `resources` del registro.

Supongamos que tiene la siguiente configuración:

- Un bucket denominado `amzn-s3-demo-bucket1` en la región `us-west-2` que contiene un objeto denominado `my-image.jpg`
- Un punto de acceso denominado `my-bucket-ap` que está asociado con `amzn-s3-demo-bucket1`
- Un Cuenta de AWS ID de `123456789012`

En el ejemplo siguiente muestra la sección `resources` de una entrada de registro de CloudTrail para la configuración anterior:

```
"resources": [
        {"type": "AWS::S3::Object",
            "ARN": "arn:aws:s3:::amzn-s3-demo-bucket1/my-image.jpg"
        },
        {"accountId": "123456789012",
            "type": "AWS::S3::Bucket",
            "ARN": "arn:aws:s3:::amzn-s3-demo-bucket1"
        },
        {"accountId": "123456789012",
            "type": "AWS::S3::AccessPoint",
            "ARN": "arn:aws:s3:us-west-2:123456789012:accesspoint/my-bucket-ap"
```

```
        }
    ]
```

Para obtener más información acerca de los registros de acceso al servidor de S3, consulte Registro de solicitudes con registro de acceso al servidor. Para obtener más información sobre AWS CloudTrail, consulte ¿Qué es AWS CloudTrail? en la Guía del usuario de AWS CloudTrail.

Uso de puntos de acceso de Amazon S3 con la consola de Amazon S3

En esta sección, se explica cómo administrar y utilizar los puntos de acceso de Amazon S3 mediante la AWS Management Console. Antes de comenzar, vaya a la página de detalles del punto de acceso que desea administrar o utilizar, tal como se describe en el procedimiento siguiente.

Temas

- Listado de puntos de acceso para la cuenta
- Listado de puntos de acceso para un bucket
- Visualización de detalles de configuración de un punto de acceso
- Uso de un punto de acceso
- Visualización de la configuración de acceso público de bloques para un punto de acceso
- Edición de una política de punto de acceso
- Eliminar un punto de acceso

Listado de puntos de acceso para la cuenta

Enumerar todos los puntos de acceso creados en su Cuenta de AWS

1. Inicie sesión AWS Management Console Management Console y abra la consola de Amazon S3 en https://console.aws.amazon.com/s3/.
2. En la barra de navegación de la parte superior de la página, elija el nombre de la Región de AWS que aparece. A continuación, elija la región para la que desee enumerar los puntos de acceso.
3. En el panel de navegación del lado izquierdo de la consola, elija Access points (Puntos de acceso).
4. En la página de access points (puntos de acceso), en access points (puntos de acceso), consulte los puntos de acceso de su Región de AWS.

5. (Opcional) Busque puntos de acceso por nombre escribiendo para ello un nombre en el campo de texto situado junto al menú desplegable Region (Región).
6. Elija el nombre del punto de acceso que desea administrar o utilizar.

Listado de puntos de acceso para un bucket

Enumerar todos los puntos de acceso en su Cuenta de AWS para un único bucket

1. Inicie sesión AWS Management Console Management Console y abra la consola de Amazon S3 en https://console.aws.amazon.com/s3/.
2. En la barra de navegación de la parte superior de la página, elija el nombre de la Región de AWS que aparece y, a continuación, elija la región para la que desea enumerar los puntos de acceso.
3. En el panel de navegación del lado izquierdo de la consola, elija Buckets (Buckets).
4. En la página Buckets (Buckets), seleccione el nombre del bucket cuyos puntos de acceso desea listar.
5. En la página de detalles del bucket, elija la pestaña Access points (Puntos de acceso).
6. Elija el nombre del punto de acceso que desea administrar o utilizar.

Visualización de detalles de configuración de un punto de acceso

1. Acceda a la página de detalles del punto de acceso cuyos detalles desea ver tal y como se describe en Listado de puntos de acceso para la cuenta.
2. En Access point overview (Información general del punto de acceso), consulte los detalles de configuración y las propiedades del punto de acceso seleccionado.

Uso de un punto de acceso

1. Acceda a la página de detalles del punto de acceso que desee utilizar tal y como se describe en Listado de puntos de acceso para la cuenta.
2. En la ficha Objects (Objetos), elija el nombre de un objeto u objetos a los que desea acceder a través del punto de acceso. En las páginas de funcionamiento de objetos, la consola muestra una etiqueta encima del nombre del bucket, en la que aparece el punto de acceso que está utilizando actualmente. Mientras utiliza el punto de acceso, solo puede realizar las operaciones con objetos permitidas por los permisos de ese punto de acceso.

> **Note**
>
> - La vista de consola siempre muestra todos los objetos del bucket. El uso de un punto de acceso como se describe en este procedimiento restringe las operaciones que puede realizar en esos objetos, pero no la posibilidad de ver si existen en el bucket.
> - S3 Management Console no admite el uso de puntos de acceso de Virtual Private Cloud (VPC) para acceder a los recursos del bucket. Para acceder a los recursos del bucket desde un punto de acceso de VPC, utilice la AWS CLI, los SDK de AWS o las API de REST de Amazon S3.

Visualización de la configuración de acceso público de bloques para un punto de acceso

1. Acceda a la página de detalles del punto de acceso cuya configuración desea ver, tal y como se describe en Listado de puntos de acceso para la cuenta.
2. Elija Permissions (Permisos).
3. En Access point policy (Política de punto de acceso), revise la configuración de bloqueo de acceso público para el punto de acceso.

> **Note**
>
> No puede cambiar la configuración de bloqueo de acceso público de un punto de acceso después de crear el punto de acceso.

Edición de una política de punto de acceso

1. Acceda a la página de detalles del punto de acceso cuya política desea editar, tal y como se describe en Listado de puntos de acceso para la cuenta.
2. Elija Permissions.
3. En Access point policy (Política de punto de acceso), elija Edit (Editar).
4. Escriba la política de punto de acceso en el campo de texto. La consola muestra automáticamente el nombre de recurso de Amazon (ARN) para el punto de acceso, que puede utilizar en la política.

Eliminar un punto de acceso

1. Vaya a la lista de puntos de acceso de la cuenta o de un bucket específico, tal y como se describe en Listado de puntos de acceso para la cuenta.
2. Seleccione el botón de opción situado junto al nombre del punto de acceso que desea eliminar.
3. Elija Delete (Eliminar).
4. Confirme que desea eliminar el punto de acceso escribiendo su nombre en el campo de texto que aparece y elija Delete (Eliminar).

Uso de un alias de estilo de bucket para su punto de acceso de bucket de S3

Al crear un punto de acceso, Amazon S3 genera de forma automática un alias que puede utilizar en lugar de un nombre de bucket de Amazon S3 para el acceso a datos. Puede utilizar este alias de punto de acceso en lugar de un nombre de recurso de Amazon (ARN) para operaciones de plano de datos de punto de acceso. Para obtener una lista de las operaciones, consulte Compatibilidad del punto de acceso con servicios de AWS.

A continuación se muestra un ejemplo de ARN y un alias de punto de acceso para un punto de acceso llamado *my-access-point*.

- ARN: `arn:aws:s3:`*region*`:`*account-id*`:accesspoint/`*my-access-point*
- Alias de punto de acceso: *my-access-point*`-`*hrzrlukc5m36ft7okagglf3gmwluquse1b*`-s3alias`

Para obtener más información acerca de los ARN, consulte Nombres de recurso de Amazon (ARN) en la Referencia general de AWS.

Nombres de alias de punto de acceso

Se crea un nombre de alias de punto de acceso en el mismo espacio de nombres que un bucket de Amazon S3. Este nombre de alias se genera de forma automática y no se puede cambiar. Un nombre de alias de punto de acceso cumple con todos los requisitos de un nombre de bucket válido de Amazon S3 y consta de las siguientes partes:

access point prefix`-`*metadata*`-s3alias`

> **Note**
>
> El sufijo `-s3alias` está reservado para los nombres de alias de punto de acceso y no se puede utilizar para los nombres de punto de acceso o bucket. Para obtener más información acerca de las reglas de nomenclatura del bucket de Amazon S3, consulte Reglas de nomenclatura de buckets.

Casos de uso y limitaciones de alias de punto de acceso

Al adoptar puntos de acceso, puede utilizar nombres de alias de puntos de acceso sin tener que hacer cambios exhaustivos en el código.

Al crear un punto de acceso, Amazon S3 genera de forma automática un nombre de alias de punto de acceso, como se muestra en el siguiente ejemplo. Para ejecutar este comando, sustituya los *`user input placeholders`* con su propia información.

```
aws s3control create-access-point --bucket amzn-s3-demo-bucket1 --name my-access-point
 --account-id 111122223333
{
    "AccessPointArn":
    "arn:aws:s3:region:111122223333:accesspoint/my-access-point",
    "Alias": "my-access-point-aqfqprnstn7aefdfbarligizwgyfouse1a-s3alias"
}
```

Puede utilizar este nombre de alias de punto de acceso en lugar de un nombre de bucket de Amazon S3 en cualquier operación de plano de datos. Para obtener una lista de las operaciones, consulte Compatibilidad del punto de acceso con servicios de AWS.

El siguiente ejemplo de AWS CLI del comando `get-object` utiliza el alias del punto de acceso del bucket para devolver información sobre el objeto especificado. Para ejecutar este comando, sustituya los *`user input placeholders`* con su propia información.

```
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": {}
}
```

Limitaciones

- Los clientes no pueden configurar los alias.
- Los alias no se pueden eliminar, modificar ni deshabillitar en un punto de acceso.
- Puede utilizar este nombre de alias de punto de acceso en lugar de un nombre de bucket de Amazon S3 en algunas operaciones de plano de datos. Para obtener una lista de las operaciones, consulte Compatibilidad de los puntos de acceso con las operaciones de S3.
- No puede utilizar un nombre de alias de punto de acceso para operaciones de plano de control de Amazon S3. Para obtener una lista de operaciones de plano de control de Amazon S3, consulte Control de Amazon S3 en la referencia de la API de Amazon Simple Storage Service.
- No puede usar los alias de los puntos de acceso de S3 como origen o destino para las operaciones de Mover en la consola de Amazon S3.
- Los alias no se pueden usar en las políticas de AWS Identity and Access Management (IAM).
- Los alias no se pueden usar como destino de registro para los registros de acceso al servidor de S3.
- Los alias no se pueden usar como destino de registro para los registros de AWS CloudTrail.
- Amazon SageMaker GroundTruth no admite los alias de punto de acceso.

Usar puntos de acceso con operaciones compatibles con Amazon S3

Los siguientes ejemplos muestran cómo utilizar puntos de acceso con operaciones compatibles en Amazon S3.

Temas

- Compatibilidad del punto de acceso con servicios de AWS
- Compatibilidad de los puntos de acceso con las operaciones de S3
- Solicitar un objeto a través de un punto de acceso
- Carga de un objeto mediante un alias de punto de acceso

- Eliminar un objeto a través de un punto de acceso
- Enumeración de objetos mediante un alias de punto de acceso
- Agregar un conjunto de etiquetas a un objeto a través de un punto de acceso
- Conceder permisos de acceso a través de un punto de acceso mediante una ACL

Compatibilidad del punto de acceso con servicios de AWS

Los alias de puntos de acceso de Amazon S3 permiten que cualquier aplicación que requiera un nombre de bucket de S3 utilice un punto de acceso fácilmente. Puede usar alias de punto de acceso de S3 donde utilice nombres de bucket de S3 para acceder a los datos en S3. Para obtener más información, consulte Casos de uso y limitaciones de alias de punto de acceso.

Compatibilidad de los puntos de acceso con las operaciones de S3

Puede utilizar puntos de acceso para acceder a un bucket mediante el siguiente subconjunto de API de Amazon S3. Todas las operaciones enumeradas a continuación aceptan ARN de puntos de acceso o alias de puntos de acceso:

Operaciones de S3

- `AbortMultipartUpload`
- `CompleteMultipartUpload`
- `CopyObject` (solo copias de la misma región)
- `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 (solo copias de la misma región)

Solicitar un objeto a través de un punto de acceso

En el ejemplo siguiente se muestra cómo solicitar el objeto `my-image.jpg` a través del punto de acceso `prod` que es propiedad del ID de cuenta `123456789012` de la región `us-west-2`; a continuación, guarda el archivo descargado como `download.jpg`.

AWS CLI

```
aws s3api get-object --key my-image.jpg --bucket arn:aws:s3:us-
west-2:123456789012:accesspoint/prod download.jpg
```

Carga de un objeto mediante un alias de punto de acceso

En el ejemplo siguiente, se carga el objeto `my-image.jpg` mediante el alias de punto de acceso `my-access-point-hrzrlukc5m36ft7okagglf3gmwluquse1b-s3alias` propiedad del ID de cuenta `123456789012` en la región `us-west-2`.

AWS CLI

```
aws s3api put-object --bucket my-access-point-hrzrlukc5m36ft7okagglf3gmwluquse1b-
s3alias --key my-image.jpg --body my-image.jpg
```

Eliminar un objeto a través de un punto de acceso

En el ejemplo siguiente se muestra cómo eliminar el objeto `my-image.jpg` a través del punto de acceso `prod` que es propiedad del ID de cuenta `123456789012` de la región `us-west-2`.

AWS CLI

```
aws s3api delete-object --bucket arn:aws:s3:us-west-2:123456789012:accesspoint/prod
 --key my-image.jpg
```

Enumeración de objetos mediante un alias de punto de acceso

En el ejemplo siguiente, se enumeran los objetos mediante el alias de punto de acceso `my-access-point-hrzrlukc5m36ft7okagglf3gmwluquse1b-s3alias` propiedad del ID de cuenta `123456789012` en la región `us-west-2`.

AWS CLI

```
aws s3api list-objects-v2 --bucket my-access-point-
hrzrlukc5m36ft7okagglf3gmwluquse1b-s3alias
```

Agregar un conjunto de etiquetas a un objeto a través de un punto de acceso

En el ejemplo siguiente se muestra cómo agregar un conjunto de etiquetas al objeto `my-image.jpg` existente a través del punto de acceso `prod` que es propiedad del ID de cuenta `123456789012` de la región `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"}]
```

Conceder permisos de acceso a través de un punto de acceso mediante una ACL

En el ejemplo siguiente se muestra cómo aplicar una ACL a un objeto `my-image.jpg` existente a través del punto de acceso `prod` que es propiedad de ID de cuenta `123456789012` de la región `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
```

Restricciones y limitaciones de los puntos de acceso

Los umbrales de error de Amazon S3 tienen en cuenta las siguientes restricciones y limitaciones:

- Cada punto de acceso está asociado con exactamente un bucket, que debe especificar al crear el punto de acceso. Después de crear un punto de acceso, no puede asociarlo a un bucket diferente. Sin embargo, puede eliminar un punto de acceso y, a continuación, crear otro con el mismo nombre y asociar ese punto de acceso nuevo a un bucket diferente.
- Los nombres de los puntos de acceso deben cumplir ciertas condiciones. Para obtener más información sobre cómo asignar nombres a los puntos de acceso, consulte Reglas para asignar nombres a los puntos de acceso de Amazon S3.
- Después de crear un punto de acceso, no puede cambiar su configuración de nube privada virtual (VPC).
- Las políticas de punto de acceso tienen un límite de tamaño de 20 KB.
- Puede crear un máximo de 10 000 puntos de acceso por Cuenta de AWS y región. Si necesita más de 10 000 puntos de acceso para una sola cuenta en una sola región, puede solicitar un aumento de la cuota de servicio. Para obtener más información sobre las cuotas de servicio y solicitar un aumento, consulte Cuotas de servicio de AWS en la Referencia general de AWS.
- En Regiones de AWS donde tenga más de 1000 puntos de acceso, no puede buscar un punto de acceso por nombre en la consola de Amazon S3.

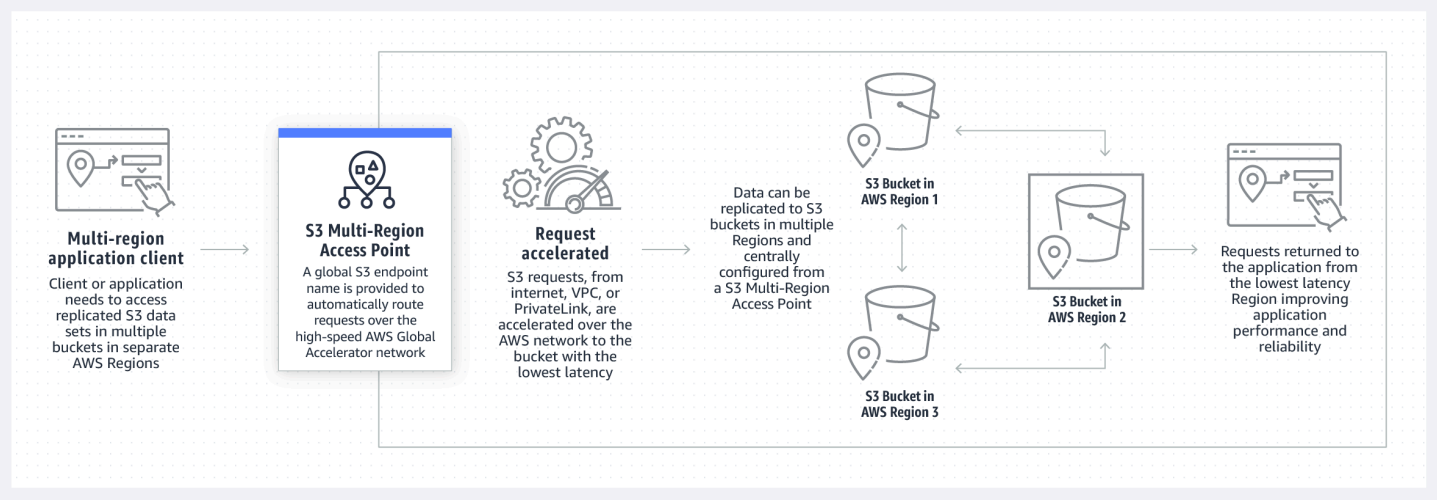
- No puede usar un punto de acceso como destino para la reproducción de S3. Para obtener más información acerca de la replicación, consulte Información general de la replicación de objetos.
- No puede usar los alias de los puntos de acceso S3 como origen o destino para las operaciones de Mover en la consola de Amazon S3.
- Puede abordar los puntos de acceso solo mediante URL de tipo de host virtual. Para obtener más información acerca del direccionamiento de tipo de host virtual, consulte Acceso y publicación de un bucket de Amazon S3.
- Las operaciones de la API que controlan la funcionalidad de los puntos de acceso (por ejemplo, `PutAccessPoint` y `GetAccessPointPolicy`) no admiten llamadas entre cuentas.
- Debe usar AWS Signature Version 4 cuando realice solicitudes a un punto de acceso mediante las API de REST. Para obtener más información sobre las solicitudes de autenticación, consulte Autenticación de solicitudes (AWS Signature Version 4) en la Referencia de la API de Amazon Simple Storage Service.
- Los puntos de acceso solo admiten solicitudes a través de HTTPS. Amazon S3 responderá automáticamente con una redirección HTTP a cualquier solicitud realizada a través de HTTP, para actualizar la solicitud a HTTPS.
- Los puntos de acceso no admiten el acceso anónimo.
- Los puntos de acceso entre cuentas no conceden acceso a los datos hasta que el propietario del bucket le conceda los permisos. El propietario del bucket siempre retiene el control máximo sobre los datos y debe actualizar la política del bucket para autorizar las solicitudes desde el punto de acceso entre cuentas. Para ver un ejemplo de política de bucket, consulte Configurar las políticas de IAM para el uso de puntos de acceso
- Al ver un punto de acceso entre cuentas en la consola de Amazon S3, la columna Acceso muestra Desconocido. La consola de Amazon S3 no puede determinar si se ha concedido acceso público al bucket y a los objetos asociados. A menos que necesite una configuración pública para un caso de uso específico, le recomendamos que usted y el propietario del bucket bloqueen todo el acceso público al punto de acceso y al bucket. Para obtener más información, consulte Bloquear el acceso público a su almacenamiento de Amazon S3.

Puntos de acceso de varias regiones de Amazon S3

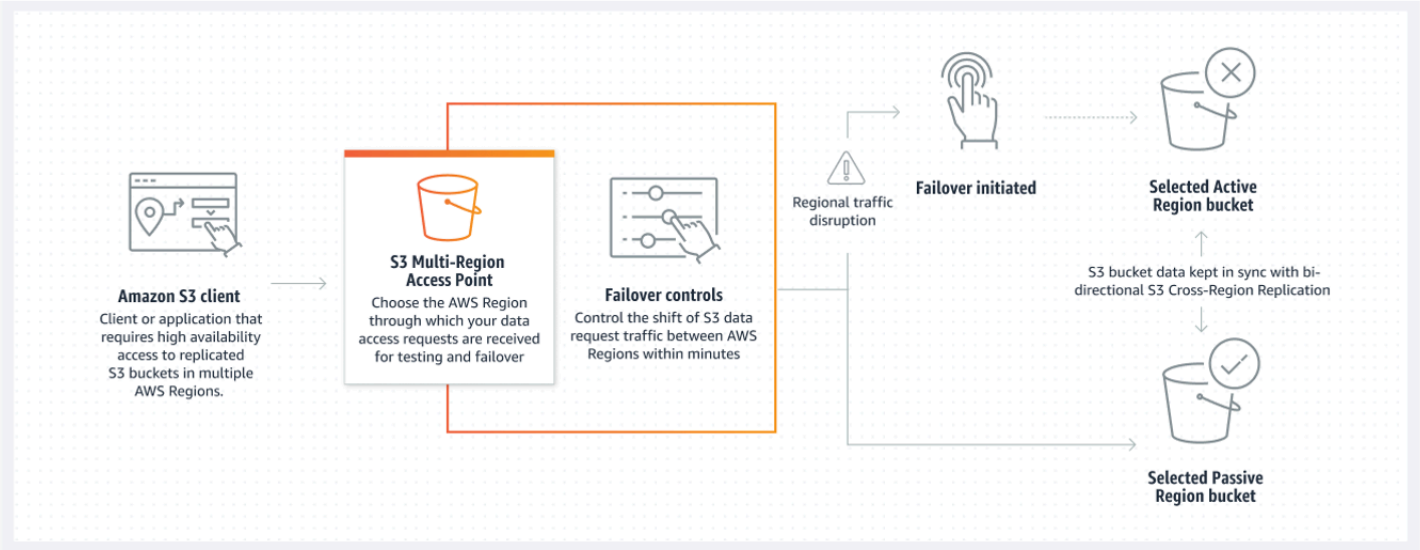
Los puntos de acceso de varias regiones de Amazon S3 proporcionan un punto de conexión global que las aplicaciones pueden utilizar para satisfacer las solicitudes de los buckets de S3 ubicados en varias Regiones de AWS. Puede utilizar puntos de acceso multirregional para crear aplicaciones de varias regiones con la misma arquitectura que se utiliza en una sola región y, a continuación, ejecutar esas aplicaciones en cualquier parte del mundo. En lugar de enviar solicitudes a través de Internet público congestionado, los puntos de acceso de varias regiones proporcionan resistencia de red integrada con la aceleración de las solicitudes basadas en Internet a Amazon S3. Las solicitudes de aplicación realizadas a un punto de conexión global de punto de acceso de varias regiones usan AWS Global Accelerator para enrutar automáticamente a través de la red global de AWS hacia el bucket de S3 más próximo con un estado de enrutamiento activo.

Cuando se crea un punto de acceso multirregional, se especifica un conjunto de Regiones de AWS en las que se desea almacenar los datos que se servirán a través de dicho punto de acceso multirregional. Puede utilizar la Replicación entre regiones (CRR) de S3 para sincronizar los datos entre los buckets de esas regiones. A continuación, puede solicitar o escribir datos a través del punto de enlace global del punto de acceso de varias regiones. Amazon S3 sirve solicitudes automáticamente al conjunto de datos replicado desde la región más cercana disponible. Los puntos de acceso de varias regiones también son compatibles con las aplicaciones que se ejecutan en las nubes privadas virtuales (VPC) de Amazon, incluidas aquellas que usan AWS PrivateLink para Amazon S3.

A continuación, se muestra una representación gráfica de un punto de acceso de varias regiones de Amazon S3 en una configuración activa-activa. El gráfico muestra cómo las solicitudes de Amazon S3 se enrutan automáticamente a los buckets de la Región de AWS activa más cercana.



La siguiente imagen es una representación gráfica de un punto de acceso de varias regiones de Amazon S3 en una configuración activa-pasiva. El gráfico muestra cómo puede controlar el tráfico de acceso a datos de Amazon S3 para realizar una conmutación por error entre Regiones de AWS activas y pasivas.



Para obtener más información acerca de cómo utilizar los puntos de acceso de varias regiones, consulte Tutorial: Introducción a los puntos de acceso de varias regiones de Amazon S3.

Temas

- Creación de puntos de acceso de varias regiones
- Configuración de un punto de acceso de varias regiones para usarlo con AWS PrivateLink
- Realización de solicitudes mediante un punto de acceso multirregional

Creación de puntos de acceso de varias regiones

Para crear un punto de acceso de varias regiones en Amazon S3, haga lo siguiente:

- Especifique el nombre del punto de acceso de varias regiones.
- Elija un bucket en cada Región de AWS a la que quiere dirigir solicitudes para el punto de acceso de varias regiones.
- Configure el bloqueo de acceso público de Amazon S3 para el punto de acceso de varias regiones.

Usted proporciona toda esta información en una solicitud de creación, que Amazon S3 procesa de forma asíncrona. Amazon S3 proporciona un token que puede utilizar para monitorear el estado de la solicitud de creación asíncrona.

Asegúrese de resolver advertencias de seguridad, errores, advertencias generales y sugerencias de AWS Identity and Access Management Access Analyzer antes de guardar la política. IAM Access Analyzer ejecuta verificaciones de política para validarla contra la Gramática de la política de IAM y las prácticas recomendadas. Estas verificaciones generan hallazgos y proporcionan recomendaciones procesables para ayudarlo a crear políticas funcionales y que se ajustan a las prácticas recomendadas de seguridad. Para obtener más información sobre la validación de políticas mediante IAM Access Analyzer, consulte Validación de políticas de IAM Access Analyzer en la Guía del usuario de IAM. Para ver una lista de advertencias, errores y sugerencias que devuelve IAM Access Analyzer, consulte Referencia de verificación de políticas de IAM Access Analyzer.

Cuando se utiliza la API, la solicitud para crear un punto de acceso de varias regiones es asincrónica. Cuando envía una solicitud para crear un punto de acceso de varias regiones, Amazon S3 autoriza la solicitud de forma sincrónica. Luego, devuelve inmediatamente un token que puede utilizar para realizar un seguimiento del progreso de la solicitud de creación. Para obtener más información sobre el seguimiento de solicitudes asincrónicas para crear y administrar puntos de acceso de varias regiones, consulte Uso de puntos de acceso de varias regiones con operaciones API admitidas.

Después de crear el punto de acceso de varias regiones, puede crear una política de control de acceso para él. Cada punto de acceso de varias regiones puede tener una política asociada. Una política de punto de acceso de varias regiones es una política basada en recursos que permite limitar el uso del punto de acceso de varias regiones en función del recurso, del usuario o de otras condiciones.

> **Note**
>
> Para que una aplicación o un usuario puedan acceder a un objeto a través de un punto de acceso de varias regiones, las dos políticas siguientes deben permitir la solicitud:
>
> - La política de acceso para el punto de acceso de varias regiones
> - La política de acceso para el bucket subyacente que contiene el objeto
>
> Cuando las dos políticas son diferentes, la política más restrictiva tiene prioridad.

Para simplificar la administración de permisos para los puntos de acceso de varias regiones, puede delegar el control de acceso del bucket al punto de acceso de varias regiones. Para obtener más información, consulte the section called “Ejemplos de política de punto de acceso multirregional”.

El uso de un bucket con un punto de acceso de varias regiones no cambia el comportamiento del bucket cuando se accede al él a través del nombre del bucket existente o del nombre de recurso de Amazon (ARN). Todas las operaciones existentes respecto al bucket continuarán funcionando como antes. Las restricciones que se incluyen en una política de punto de acceso de varias regiones solo se aplican a las solicitudes realizadas a través de ese punto de acceso de varias regiones.

Puede actualizar la política para un punto de acceso de varias regiones después de crearla, pero no puede eliminarla. Sin embargo, puede actualizar la política de punto de acceso de varias regiones para denegar todos los permisos.

Temas

Reglas para asignar nombres a los puntos de acceso de varias regiones de Amazon S3

Cuando crea un punto de acceso de varias regiones, le asigna un nombre, que es una cadena que elige. Después de crearlo, no se puede cambiar el nombre del punto de acceso de varias regiones. El nombre debe ser único en su Cuenta de AWS y debe cumplir con los requisitos para la designación de nombres enumerados en Restricciones y limitaciones de puntos de acceso de varias regiones. Para ayudarle a identificar el punto de acceso de varias regiones, utilice un nombre que sea significativo para usted, para la organización o que refleje el escenario.

Este nombre se utiliza cuando se invocan operaciones de administración de puntos de acceso de varias regiones, como `GetMultiRegionAccessPoint` y `PutMultiRegionAccessPointPolicy`. El nombre no se utiliza para enviar solicitudes al punto de acceso de varias regiones y no necesita estar expuesto a clientes que realizan solicitudes mediante el punto de acceso de varias regiones.

Cuando Amazon S3 crea un punto de acceso de varias regiones, le asigna automáticamente un alias. Este alias es una cadena alfanumérica única que termina en `.mrap`. El alias se utiliza para construir el nombre de host y el nombre de recurso de Amazon (ARN) para un punto de acceso de varias regiones. El nombre completo también se basa en el alias del punto de acceso de varias regiones.

No se puede determinar el nombre de un punto de acceso de varias regiones a partir de su alias, por lo que puede revelar un alias sin riesgo de exponer el nombre, el propósito o el propietario del punto de acceso de varias regiones. Amazon S3 selecciona el alias para cada nuevo punto de acceso de varias regiones y el alias no se puede cambiar. Para obtener más información acerca de la dirección de un punto de acceso de varias regiones, consulte Realización de solicitudes mediante un punto de acceso multirregional.

Los alias de punto de acceso de varias regiones son únicos a lo largo del tiempo y no se basan en el nombre o la configuración de un punto de acceso de varias regiones. Si crea un punto de acceso de varias regiones y, a continuación, lo elimina y crea otro con el mismo nombre y configuración, el segundo punto de acceso de varias regiones tendrá un alias diferente al primero. Los nuevos puntos deacceso de varias regiones nunca pueden tener el mismo alias que un punto de acceso de varias regiones anterior.

Reglas para elegir buckets para puntos de acceso de varias regiones de Amazon S3

Cada punto de acceso de varias regiones está asociado a las regiones en las que desea satisfacer las solicitudes. El punto de acceso de varias regiones debe estar asociado a exactamente un bucket en cada una de esas regiones. Especifique el nombre de cada bucket en la solicitud para crear el punto de acceso de varias regiones. Los buckets que admiten el punto de acceso de varias regiones pueden estar en la misma Cuenta de AWS que posee el punto de acceso de varias regiones o pueden estar en otra Cuentas de AWS.

Un solo bucket puede ser utilizado por varios puntos de acceso de varias regiones.

Important

- Puede especificar los buckets asociados a un punto de acceso de varias regiones solo en el momento en que lo cree. Después de crearlo, no puede agregar, modificar ni eliminar buckets de la configuración del punto de acceso de varias regiones. Para cambiar los buckets, debe eliminar todo el punto de acceso de varias regiones y crear uno nuevo.
- No se puede eliminar un bucket que forme parte de un punto de acceso de varias regiones. Si desea eliminar un bucket vinculado a un punto de acceso de varias regiones, elimine primero el punto de acceso de varias regiones.
- Si agrega un bucket que pertenece a otra cuenta a un punto de acceso de varias regiones, el propietario del bucket también debe actualizar la política de bucket para conceder permisos de acceso al punto de acceso de varias regiones. De lo contrario, el punto de acceso de varias regiones no podrá recuperar los datos de ese bucket. Para ver políticas que muestran cómo conceder dicho acceso, consulte Ejemplos de política de punto de acceso multirregional.
- No todas las regiones admiten puntos de acceso de varias regiones. Para ver las lista de regiones admitidas, consulte Restricciones y limitaciones de puntos de acceso de varias regiones.

Puede crear reglas de replicación para sincronizar datos entre buckets. Estas reglas le permiten copiar automáticamente los datos de los buckets de origen a los buckets de destino. Tener buckets conectados a un punto de acceso de varias regiones no afecta el funcionamiento de la replicación. La configuración de replicación con puntos de acceso de varias regiones se describe en una sección posterior.

Important

Cuando realiza una solicitud a un punto de acceso de varias regiones, este no conoce el contenido de datos de los buckets del punto de acceso de varias regiones. Por lo tanto, es posible que el bucket que reciba la solicitud no contenga los datos solicitados. Para crear conjuntos de datos coherentes en los buckets de Amazon S3 asociados con un punto de acceso de varias regiones, le recomendamos que configure la replicación entre regiones (CRR) de S3. Para obtener más información, consulte Configuración de la replicación de bucket para utilizarla con puntos de acceso de varias regiones.

Crear un punto de acceso de varias regiones de Amazon S3

En los siguientes ejemplos se muestra cómo crear un punto de acceso de varias regiones mediante la consola de Amazon S3.

Uso de la consola de S3

Para crear un punto de acceso de varias regiones

1. Inicie sesión AWS Management Console Management Console y abra la consola de Amazon S3 en https://console.aws.amazon.com/s3/.
2. En el panel de navegación izquierdo, elija Puntos de acceso de varias regiones.
3. Elija Crear puntos de acceso de varias regiones para empezar a crear el punto de acceso de varias regiones.
4. En la página Punto de acceso de varias regiones, proporcione un nombre para el punto de acceso de varias regiones en el campo Nombre del punto de acceso de varias regiones.
5. Seleccione los buckets que se asociarán a este punto de acceso de varias regiones. Puede elegir los buckets que están en su cuenta o los buckets de otras cuentas.

> **Note**
>
> Debe añadir al menos un bucket de su cuenta o de otras cuentas. Además, debe tener en cuenta que los puntos de acceso de varias regiones solo admiten un bucket por Región de AWS. Por lo tanto, no puede añadir dos buckets de la misma región. No se admiten Regiones de AWS deshabilitadas de forma predeterminada.

   - Para añadir un bucket de su cuenta, seleccione Agregar buckets. Se muestra una lista de los buckets de la cuenta. Puede buscar el bucket por nombre o clasificar los nombres de los buckets por orden alfabético.
   - Para agregar un bucket de otra cuenta, seleccione Agregar bucket desde otras cuentas. Debe saber el nombre del bucket y el ID de Cuenta de AWS exactos, pues no puede buscar buckets en otras cuentas.

> **Note**
>
> Debes introducir un ID Cuenta de AWS y un nombre de bucket válidos. El bucket también debe estar en una región compatible; de lo contrario, se producirá un error al intentar crear el punto de acceso de varias regiones. Para ver la lista de regiones que admiten puntos de acceso de varias regiones, consulte Restricciones y limitaciones de puntos de acceso de varias regiones.

6. (Opcional) Si tiene que eliminar un bucket que ha añadido, seleccione Eliminar.

> **Note**
>
> No puede agregar ni eliminar buckets a este punto de acceso de varias regiones después de crearlo.

7. En Configuración de bloqueo del acceso público a este punto de acceso de varias regiones, seleccione la configuración de bloqueo de acceso público que desee aplicar al punto de acceso de varias regiones. Todas las configuraciones de bloqueo de acceso público están habilitadas de forma predeterminada para los nuevos puntos de acceso de varias regiones. Le recomendamos que deje todas las configuraciones habilitadas a menos que sepa que tiene una necesidad específica de desactivar cualquiera de ellas.

> **Note**
>
> No puede cambiar la configuración de Bloquear acceso público para un punto de acceso de varias regiones una vez creado. Por lo tanto, si va a bloquear el acceso público, asegúrese de que las aplicaciones funcionen correctamente sin acceso público antes de crear un punto de acceso multirregional.

8. Seleccione Crear punto de acceso de varias regiones.

> **Important**
>
> Si agrega un bucket que pertenece a otra cuenta a un punto de acceso de varias regiones, el propietario del bucket también debe actualizar la política de bucket para conceder permisos de acceso al punto de acceso de varias regiones. De lo contrario, el punto de acceso de varias regiones no podrá recuperar los datos de ese bucket. Para ver políticas que

muestran cómo conceder dicho acceso, consulte Ejemplos de política de punto de acceso multirregional.

Uso de la AWS CLI

Puede utilizar AWS CLI para crear un punto de acceso de varias regiones. Cuando cree el punto de acceso de varias regiones, debe proporcionar todos los buckets que admitirá. No ouede agregar buckets al punto de acceso de varias regiones una vez creado.

En el ejemplo siguiente se crea un punto de acceso de varias regiones con dos buckets mediante AWS CLI. Para utilizar este comando de ejemplo, sustituya *user input placeholders* por su propia información.

```
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": "amzn-s3-demo-bucket1" },
            { "Bucket": "amzn-s3-demo-bucket2" }
        ]
    }' --region us-west-2
```

Bloqueo del acceso público con puntos de acceso de varias regiones de Amazon S3

Cada punto de acceso de varias regiones tiene configuraciones distintas para Bloquear acceso público de Amazon S3. Esta configuración funciona junto con la configuración de Bloquear acceso público para la Cuenta de AWS que posee tanto el punto de acceso de varias regiones como los buckets subyacentes.

Cuando Amazon S3 autoriza una solicitud, aplica la combinación más restrictiva de esta configuración. Si la configuración de Bloquear acceso público de cualquiera de estos recursos (el punto de acceso de varias regiones, el bucket subyacente o la cuenta del propietario del bucket) bloquea el acceso a la acción o recurso solicitados, Amazon S3 rechaza la solicitud.

Recomendamos dejar todas estas configuraciones de Bloquear acceso público habilitadas, a menos que sepa que tiene una necesidad específica de desactivar cualquiera de ellas. Todas las configuraciones de bloqueo de acceso público están habilitadas de forma predeterminada para los puntos de acceso de varias regiones. Si Bloquear acceso público está habilitado, el punto de acceso de varias regiones no podrá aceptar solicitudes basadas en internet.

> **Important**
>
> No puede cambiar la configuración de Bloquear acceso público después de que se cree el punto de acceso de varias regiones.

Para obtener más información sobre el bloqueo de acceso público de Amazon S3, consulte Bloquear el acceso público a su almacenamiento de Amazon S3.

Visualización de los de talles de configuración de los puntos de acceso de varias regiones de Amazon S3

En los siguientes ejemplos se muestra cómo visualizar los detalles de configuración de un punto de acceso de varias regiones mediante la consola de Amazon S3.

Uso de la consola de S3

Para crear un punto de acceso de varias regiones

1. Inicie sesión AWS Management Console Management Console y abra la consola de Amazon S3 en https://console.aws.amazon.com/s3/.
2. En el panel de navegación izquierdo, elija Puntos de acceso de varias regiones.
3. Elija el nombre del punto de acceso de varias regiones del que quiere ver los detalles de configuración.

    - La pestaña Propiedades muestra todos los buckets que están asociados a su punto de acceso de varias regiones, la fecha de creación, el nombre de recurso de Amazon (ARN) y el alias. La columna ID Cuenta de AWS también muestra los buckets que pertenecen a cuentas externas asociadas a su punto de acceso de varias regiones.
    - La pestaña Permisos muestra la configuración de Bloquear acceso público que se aplica a los buckets asociados con este punto de acceso de varias regiones. También puede ver la política de punto de acceso de varias regiones para su punto de acceso de varias regiones,

si ha creado uno. La alerta Información de la página Permisos también muestra todos los buckets (en su cuenta y en otras cuentas) de este punto de acceso de varias regiones que tienen habilitada la configuración El acceso público está bloqueado.

- La pestaña Replicación y conmutación por error proporciona un mapa visual de los buckets asociados a su punto de acceso de varias regiones y las regiones en las que residen los buckets. Si hay buckets de otra cuenta de los que no tiene permiso para extraer datos, la región estará marcada en rojo en el mapa Resumen de replicación, lo que indica que se trata de una Región de AWS con errores al obtener el estado de replicación.

  > **Note**
  >
  > Para recuperar la información del estado de la replicación de un bucket en una cuenta externa, el propietario del bucket debe concederle el permiso `s3:GetBucketReplication` en su política de bucket.

  Esta pestaña también proporciona las métricas de replicación, las reglas de replicación y los estados de conmutación por error para las regiones que se utilizan con su punto de acceso de varias regiones.

Uso de la AWS CLI

Puede utilizar la AWS CLI para ver los detalles de configuración de un punto de acceso de varias regiones.

El siguiente ejemplo de la AWS CLI recibe la configuración actual del punto de acceso de varias regiones. Para utilizar este comando de ejemplo, sustituya *`user input placeholders`* por su propia información.

```
aws s3control get-multi-region-access-point --account-id 111122223333 --name amzn-s3-demo-bucket1
```

Eliminación de un punto de acceso de varias regiones

En el siguiente procedimiento se explica cómo eliminar un punto de acceso de varias regiones mediante la consola de Amazon S3.

La eliminación de un punto de acceso de varias regiones no elimina los buckets asociados con el punto de acceso de varias regiones, solo el punto de acceso de varias regiones en sí.

Uso de la consola de S3

Para eliminar un punto de acceso de varias regiones

1. Inicie sesión AWS Management Console Management Console y abra la consola de Amazon S3 en https://console.aws.amazon.com/s3/.
2. En el panel de navegación izquierdo, elija Puntos de acceso de varias regiones.
3. Seleccione el botón de opción situado junto al nombre de su punto de acceso de varias regiones.
4. Elija Eliminar.
5. En el cuadro de diálogo Eliminar punto de acceso de varias regiones, introduzca el nombre del bucket AWS que desea eliminar.

> **Note**
>
> Asegúrese de introducir un nombre de bucket válido. De lo contrario, se desactivará el botón Eliminar.

6. Elija Eliminar para confirmar la eliminación de un punto de acceso de varias regiones.

Uso de la AWS CLI

Puede utilizar la AWS CLI para eliminar un punto de acceso de varias regiones. Esta acción no elimina los buckets asociados con el punto de acceso de varias regiones, solo el punto de acceso de varias regiones en sí. Para utilizar este comando de ejemplo, sustituya *`user input placeholders`* por su propia información.

```
aws s3control delete-multi-region-access-point --account-id 123456789012 --details
 Name=example-multi-region-access-point-name
```

Configuración de un punto de acceso de varias regiones para usarlo con AWS PrivateLink

Puede utilizar puntos de acceso de varias regiones para dirigir el tráfico de solicitudes de Amazon S3 entre Regiones de AWS. Cada punto de acceso de varias regiones global dirige el tráfico de las solicitudes de datos de Amazon S3 desde múltiples orígenes sin que tenga que crear configuraciones de red complejas con puntos de conexión independientes. Estos orígenes del tráfico de solicitudes de datos incluyen:

- Tráfico que se origina en una nube privada virtual (VPC)
- Tráfico de los centros de datos en las instalaciones que viaja a través de AWS PrivateLink
- Tráfico de la internet pública

Si establece una conexión AWS PrivateLink con un punto de acceso de varias regiones de S3, puede dirigir las solicitudes de S3 hacia AWS o entre varias Regiones de AWS a través de una conexión privada mediante una arquitectura y una configuración de red sencillas. Si usa AWS PrivateLink, no es necesario configurar una conexión de emparejamiento de VPC.

Temas

Configuración de un punto de acceso de varias regiones para su uso con AWS PrivateLink

AWS PrivateLink le proporciona conectividad privada a Amazon S3 mediante direcciones IP privadas en su nube virtual privada (VPC). Puede aprovisionar uno o más puntos de enlace de interfaz dentro de su VPC para conectarse a los puntos de acceso de varias regiones de Amazon S3.

Puede crear puntos de enlace com.amazonaws.s3-global.accesspoint para los puntos de acceso de varias regiones a través de AWS Management Console, AWS CLI o AWS SDK. Para obtener más información acerca de cómo configurar un punto de enlace de interfaz para el punto de acceso de varias regiones, consulte Puntos de enlace de la VPC de tipo interfaz en la Guía del usuario de VPC.

Para realizar solicitudes a un punto de acceso de varias regiones a través de puntos de enlace de interfaz, siga estos pasos para configurar la VPC y el punto de acceso de varias regiones.

Para configurar un punto de acceso de varias regiones para utilizarlo con AWS PrivateLink

1. Cree o tenga un punto de enlace de VPC adecuado que pueda conectarse a puntos de acceso de varias regiones. Para obtener más información sobre puntos de enlace de la VPC, consulte Puntos de enlace de la VPC de tipo interfaz en la Guía del usuario de la VPC.

   > **Important**
   >
   > Asegúrese de crear un punto de enlace com.amazonaws.s3-global.accesspoint. Otros tipos de puntos de enlace no pueden acceder a los puntos de acceso de varias regiones.

   Después de crear este punto de enlace de VPC, todas las solicitudes de Punto de acceso de varias regiones de la VPC se dirigen a través de este punto de enlace si tiene habilitado DNS privado para el punto de enlace. Esto está habilitado de forma predeterminada.
2. Si la política de punto de acceso de varias regiones no admite conexiones desde puntos de enlace de VPC, deberá actualizarla.
3. Compruebe que las políticas de bucket individuales permitirán el acceso a los usuarios del punto de acceso de varias regiones.

Recuerde que los puntos de acceso de varias regiones funcionan dirigiendo las solicitudes a los buckets, no cumpliendo las solicitudes por sí mismos. Es importante recordar esto porque el originador de la solicitud debe tener permisos para el punto de acceso de varias regiones y tener permiso para acceder a los buckets individuales en el punto de acceso de varias regiones. De lo contrario, la solicitud podría dirigirse a un bucket en el que el iniciador no tenga permisos para cumplir con la solicitud. Un punto de acceso de varias regiones y los buckets asociados pueden pertenecer a la misma cuenta o a otra cuenta de AWS. Sin embargo, las VPC de diferentes cuentas pueden utilizar un punto de acceso de varias regiones si los permisos están configurados correctamente.

Debido a esto, la política de punto de enlace de VPC debe permitir el acceso tanto al punto de acceso de varias regiones como a cada bucket subyacente que desee poder satisfacer las solicitudes. Por ejemplo, suponga que tiene un punto de acceso multirregional con el alias `mfzwi23gnjvgw.mrap`. Está respaldado por los buckets `amzn-s3-demo-bucket1` y `amzn-s3-demo-bucket2`, todos propiedad de la cuenta de AWS `123456789012`. En este caso, la siguiente política de punto de conexión de VPC permitiría solicitudes `GetObject` de la VPC hechas a `mfzwi23gnjvgw.mrap` para que cualquiera de los buckets de respaldo las responda.

```
{
    "Version": "2012-10-17",
    "Statement": [
    {
        "Sid": "Read-buckets-and-MRAP-VPCE-policy",
        "Principal": "*",
        "Action": [
            "s3:GetObject"
        ],
        "Effect": "Allow",
        "Resource": [
            "arn:aws:s3:::amzn-s3-demo-bucket1/*",
            "arn:aws:s3:::amzn-s3-demo-bucket2/*",
            "arn:aws:s3::123456789012:accesspoint/mfzwi23gnjvgw.mrap/object/*"
        ]
    }]
}
```

Como se mencionó anteriormente, también debe asegurarse de que la política de punto de acceso de varias regiones esté configurada para admitir el acceso a través de un punto de enlace de VPC. No tiene que especificar el punto de enlace de VPC que solicita acceso. La siguiente política de ejemplo concedería acceso a cualquier solicitante que intente utilizar el punto de acceso multirregional para las solicitudes de `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/*"
    }]
}
```

Y, por supuesto, cada uno de los buckets individuales necesitaría una política para admitir el acceso desde las solicitudes enviadas a través del punto de enlace de VPC. En el siguiente ejemplo de

política se concede acceso de lectura a todos los usuarios anónimos, lo que incluye solicitudes realizadas a través del punto de enlace de VPC.

```
{
    "Version":"2012-10-17",
   "Statement": [
   {
       "Sid": "Public-read",
       "Effect":"Allow",
       "Principal": "*",
       "Action": "s3:GetObject",
       "Resource": [
           "arn:aws:s3:::amzn-s3-demo-bucket1",
           "arn:aws:s3:::amzn-s3-demo-bucket2/*"]
    }]
}
```

Para obtener información acerca de cómo editar una política de punto de conexión de VPC, consulte Control del acceso a los servicios con puntos de conexión de VPC en la Guía del usuario de VPC.

Eliminación del acceso a un punto de acceso de varias regiones desde un punto de enlace de VPC

Si posee un punto de acceso de varias regiones y desea quitarle el acceso desde un punto de conexión de interfaz, debe proporcionar una nueva política de acceso al punto de acceso de varias regiones que impida el acceso a las solicitudes que lleguen a través de los puntos de conexión de VPC. Si los buckets de su punto de acceso de varias regiones admiten solicitudes a través de puntos de conexión de VPC, seguirán admitiendo estas solicitudes. Si desea evitar esa compatibilidad, también debe actualizar las políticas de los buckets. El suministro de una nueva política de acceso al punto de acceso de varias regiones solo impide el acceso al punto de acceso de varias regiones, no a los buckets subyacentes.

> **Note**
>
> No se puede eliminar una política de acceso para un punto de acceso de varias regiones. Para quitar el acceso a un punto de acceso de varias regiones, debe proporcionar una nueva política de acceso con el acceso modificado que desee.

En lugar de actualizar la política de acceso para el punto de acceso de varias regiones, puede actualizar las políticas de bucket para evitar solicitudes a través de puntos de conexión de VPC. En este caso, los usuarios aún podrían acceder al punto de acceso de varias regiones a través del punto de conexión de la VPC. Pero si la solicitud de punto de acceso de varias regiones se dirige a un bucket donde la política de buckets impide el acceso, esta solicitud generaría un mensaje de error.

Realización de solicitudes mediante un punto de acceso multirregional

Debido a otros recursos, los puntos de acceso de varias regiones de Amazon S3 incluyen nombres de recursos de Amazon (ARN). Puede usar estos ARN para dirigir las solicitudes a los puntos de acceso de varias regiones mediante la AWS Command Line Interface (AWS CLI), los SDK de AWS o la API de Amazon S3. También puede utilizar estos ARN para identificar puntos de acceso de varias regiones en las políticas de control de acceso. Un ARN de punto de acceso de varias regiones no incluye ni divulga el nombre del punto de acceso de varias regiones. Para obtener más información acerca de los ARN, consulte Nombres de recurso de Amazon (ARN) en la Referencia general de AWS.

> **Note**
>
> El alias de punto de acceso multirregional y el ARN no se pueden usar indistintamente.

Los ARN de los puntos de acceso de varias regiones utilizan el siguiente formato:

`arn:aws:s3::`*`account-id`*`:accesspoint/`*`MultiRegionAccessPoint_alias`*

A continuación, se muestran algunos ejemplos de ARN de puntos de acceso de varias regiones:

- `arn:aws:s3::123456789012:accesspoint/mfzwi23gnjvgw.mrap` representa el punto de acceso de varias regiones con el alias `mfzwi23gnjvgw.mrap`, propiedad de Cuenta de AWS `123456789012`.
- `arn:aws:s3::123456789012:accesspoint/*` representa todos los puntos de acceso de varias regiones en la cuenta `123456789012`. Este ARN coincide con todos los puntos de acceso de varias regiones de la cuenta `123456789012`, pero no coincide con ningún punto de acceso de Amazon S3 regional porque el ARN no incluye una Región de AWS. En cambio, el ARN `arn:aws:s3:us-west-2:123456789012:accesspoint/*` coincide con todos los puntos de

acceso de Amazon S3 regionales de la región `us-west-2` para la cuenta `123456789012`, pero no coincide con ningún punto de acceso de varias regiones.

Los ARN para los objetos a los que se accede a través de un punto de acceso multirregional utilizan el siguiente formato:

`arn:aws:s3::`*`account_id`*`:accesspoint/`*`MultiRegionAccessPoint_alias`*`//`*`key`*

Al igual que con los ARN de puntos de acceso de varias regiones, los ARN de los objetos a los que se accede a través de los puntos de acceso de varias regiones no incluyen una Región de AWS. Estos son algunos ejemplos.

- `arn:aws:s3::123456789012:accesspoint/mfzwi23gnjvgw.mrap//-01` representa el `-01`, al que se accede a través del punto de acceso de varas regiones con el alias `mfzwi23gnjvgw.mrap`, que es propiedad de la cuenta `123456789012`.
- `arn:aws:s3::123456789012:accesspoint/mfzwi23gnjvgw.mrap//*` representa todos los objetos a los que se puede acceder a través del punto de acceso multirregional con el alias `mfzwi23gnjvgw.mrap`, en la cuenta `123456789012`.
- `arn:aws:s3::123456789012:accesspoint/mfzwi23gnjvgw.mrap//-01/finance/*` representa todos los objetos a los que se puede acceder con el prefijo `-01/finance/` para el punto de acceso de varias regiones con el alias `mfzwi23gnjvgw.mrap`, en la cuenta `123456789012`.

Nombres de host de punto de acceso de varias regiones

Puede acceder a los datos de Amazon S3 a través de un punto de acceso multirregional utilizando el nombre de host del punto de acceso multirregional. Las solicitudes se pueden dirigir a este nombre del host desde la Internet pública. Si ha configurado una o más puertas de enlace de Internet para el punto de acceso multirregional, las solicitudes se pueden dirigir a este nombre de host desde una nube privada virtual (VPC). Para obtener más información acerca de la creación de puntos de enlace de la interfaz de la VPC para utilizar con puntos de acceso de multirregiones, consulteConfiguración de un punto de acceso de varias regiones para su uso con AWS PrivateLink.

Para realizar solicitudes a través de un punto de acceso de varias regiones desde una VPC mediante un punto de conexión de VPC, puede usar AWS PrivateLink. Si realiza solicitudes a un punto de acceso de varias regiones utilizando AWS PrivateLink, no puede usar directamente un sistema de nombres de dominio (DNS) regional específico de punto de conexión que termine

con *region*.vpce.amazonaws.com. Este nombre de host no tendrá un certificado asociado con él, por lo que no se puede usar directamente. Puede utilizar el nombre del sistema de nombres de dominio (DNS) público del punto de conexión de VPC como CNAME o destino de ALIAS. De forma alternativa, puede habilitar el sistema de nombres de dominio (DNS) privado en el punto de conexión y utilizar los nombres de sistema de nombres de dominio (DNS) *MultiRegionAccessPoint_alias*.accesspoint.s3-global.amazonaws.com del punto de acceso de varias regiones estándar, tal como se describe en esta sección.

Cuando realiza solicitudes a la API para operaciones de datos de Amazon S3 (por ejemplo, GetObject) a través de un punto de acceso de varias regiones, el nombre de host para la solicitud es el siguiente:

MultiRegionAccessPoint_alias.accesspoint.s3-global.amazonaws.com

Por ejemplo, para hacer una solicitud de GetObject a través del punto de acceso multirregional con el alias mfzwi23gnjvgw.mrap, haga una solicitud al nombre de host mfzwi23gnjvgw.mrap.accesspoint.s3-global.amazonaws.com. La parte s3-global del nombre de host indica que este nombre de host no es para una región específica.

Realizar solicitudes a través de un punto de acceso de varias regiones es similar a realizar solicitudes a través de un punto de acceso de una sola región. Sin embargo, es importante tener en cuenta las siguientes diferencias:

- Los ARN de puntos de acceso multirregional no incluyen una Región de AWS. Ellos siguen el formato arn:aws:s3::*account-id*:accesspoint/*MultiRegionAccessPoint_alias*.
- Para las solicitudes realizadas a través de las operaciones de API (estas solicitudes no requieren el uso de un ARN), los puntos de acceso de varias regiones utilizan un esquema de punto de conexión diferente. El esquema es *MultiRegionAccessPoint_alias*.accesspoint.s3-global.amazonaws.com; por ejemplo, mfzwi23gnjvgw.mrap.accesspoint.s3-global.amazonaws.com. Tenga en cuenta las diferencias en comparación con un punto de acceso de una sola región:
  - Los nombres de host de puntos de acceso de varias regiones utilizan su alias, no el nombre del punto de acceso de varias regiones.
  - Los nombres de host de puntos de acceso de varias regiones no incluyen el ID de Cuenta de AWS del propietario.
  - Los nombres de host de puntos de acceso de varias regiones no incluyen una Región de AWS.
  - Los nombres de host de puntos de acceso de varias regiones incluyen s3-global.amazonaws.com en lugar de s3.amazonaws.com.

- Las solicitudes de puntos de acceso de varias regiones se deben firmar con Signature Version 4A (SigV4A). Si utiliza los SDK de AWS, el SDK convierte automáticamente SigV4 en SigV4A. Por ello es importante comprobar que su AWS SDK sea compatible con SigV4A como implementación de firma que se utiliza para firmar las solicitudes de Región de AWS globales. Para obtener más información acerca de SigV4A, consulte Cómo firmar solicitudes de API de AWS en la Referencia general de AWS.

Puntos de acceso de varias regiones y Amazon S3 Transfer Acceleration

Amazon S3 Transfer Acceleration es una característica que permite realizar transferencias de datos a buckets. Transfer Acceleration se configura en el nivel de bucket individual. Para obtener más información acerca de Transfer Acceleration, consulte Configuración de transferencias de archivos rápidas y seguras con Amazon S3 Transfer Acceleration.

Los puntos de acceso de varias regiones usan un mecanismo de transferencia acelerada similar como Transfer Acceleration para enviar objetos grandes a través de la red de AWS. Debido a esto, no es necesario usar Transfer Acceleration al enviar solicitudes a través de un punto de acceso de varias regiones. Este mayor rendimiento de transferencia se incorpora automáticamente en el punto de acceso multirregional.

Temas

- Permisos
- Restricciones y limitaciones de puntos de acceso de varias regiones
- Enrutamiento de solicitud de punto de acceso de varias regiones
- Controles de conmutación por error de puntos de acceso de varias regiones de Amazon S3
- Configuración de la replicación de bucket para utilizarla con puntos de acceso de varias regiones
- Uso de puntos de acceso de varias regiones con operaciones API admitidas
- Monitoreo y registro de solicitudes realizadas a través de un punto de acceso de varias regiones a los recursos subyacentes

Permisos

Los puntos de acceso de varias regiones de Amazon S3 pueden simplificar el acceso a los datos para buckets de Amazon S3 en varias Regiones de AWS. Los puntos de acceso de varias regiones se llaman puntos de conexión globales que se pueden utilizar para realizar operaciones con

objetos de acceso a datos de Amazon S3, como `GetObject` y `PutObject`. Cada punto de acceso multirregional puede tener permisos y controles de red distintos para cualquier solicitud que se realice a través del punto de conexión global.

Cada punto de acceso de varias regiones también se puede aplicar a una política de acceso personalizada que funciona en conjunción con la política de bucket asociada al bucket subyacente. Para que una solicitud tenga éxito, todos los requisitos siguientes deben permitir la operación:

- La política del punto de acceso multirregional
- La política de AWS Identity and Access Management (IAM) subyacente
- La política de bucket subyacente (a la que se dirige la solicitud)

Puede configurar cualquier política de puntos de acceso de varias regiones para que solo acepte solicitudes de usuarios o grupos de IAM específicos. Para ver un ejemplo práctico, consulte el ejemplo 2 en the section called “Ejemplos de política de punto de acceso multirregional”. Para restringir el acceso a los datos de Amazon S3 a una red privada, puede configurar la política del punto de acceso de varias regiones para aceptar solo las solicitudes procedentes de una nube privada virtual (VPC).

Por ejemplo, suponga que crea una solicitud `GetObject` a través de un punto de acceso multirregional utilizando un usuario llamado `AppDataReader` en la cuenta de AWS. Para ayudar a garantizar que no se denegará la solicitud, al usuario de `AppDataReader` se le debe conceder el permiso `s3:GetObject` mediante el punto de acceso multirregional y por cada bucket subyacente el punto de acceso multirregional. `AppDataReader` no podrá recuperar datos de ningún bucket que no conceda este permiso.

> **Important**
>
> La delegación del control de acceso para un bucket a una política de punto de acceso multirregional no cambia el comportamiento del bucket cuando se accede al bucket a través del nombre de recurso de Amazon (ARN). Todas las operaciones realizadas directamente con respecto al bucket continuarán funcionando como antes. Las restricciones que se incluyen en una política de punto de acceso de varias regiones solo se aplican a las solicitudes realizadas a través de ese punto de acceso multirregional.

Administración del acceso público a un punto de acceso de varias regiones

Los puntos de acceso de varias regiones admiten configuraciones de Bloquear punto de acceso independientes para cada punto de acceso de varias regiones. Al crear un punto de acceso de varias regiones, puede especificar la configuración del bloqueo de acceso público aplicable a ese punto de acceso de varias regiones.

> **Note**
>
> Cualquier configuración de bloqueo de acceso público que esté habilitada en Configuración de bloqueo de acceso público correspondiente a esta cuenta (en su propia cuenta) o Bloquear la configuración pública para los buckets externos seguirá siendo válida incluso si la configuración independiente de bloqueo del acceso público para su punto de acceso de varias regiones está deshabilitada.

Para cualquier solicitud que se realice a través de un punto de acceso multirregional, Amazon S3 evalúa la configuración de bloqueo de acceso público para:

- El punto de acceso multirregional
- Los buckets subyacentes (incluidos los buckets externos)
- La cuenta que posee el punto de acceso de varias regiones
- La cuenta que posee los buckets subyacentes (incluidas las cuentas externas)

Si alguna de estas configuraciones indica que la solicitud debe bloquearse, Amazon S3 rechaza la solicitud. Para obtener más información acerca de la característica Bloquear acceso público en S3, consulte Bloquear el acceso público a su almacenamiento de Amazon S3.

> **Important**
>
> Todas las configuraciones de bloqueo de acceso público están habilitadas de forma predeterminada para los puntos de acceso de varias regiones. Debe deshabilitar explícitamente cualquier configuración que no desee aplicar a un punto de acceso de varias regiones.
>
> No puede cambiar la configuración de Bloquear acceso público después de que se cree el punto de acceso de varias regiones.

Visualización de la configuración de Block Public Access para un punto de acceso multirregional

Para ver la configuración de Block Public Access para un punto de acceso multirregional

1. Inicie sesión AWS Management Console Management Console y abra la consola de Amazon S3 en https://console.aws.amazon.com/s3/.
2. En el panel de navegación izquierdo, elija Puntos de acceso de varias regiones.
3. Elija el nombre del punto de acceso de varias regiones que desee revisar.
4. Elija la pestaña Permisos.
5. En Block Public Access settings for this Multi-Region Access Point (Configuración de bloqueo del acceso público a este punto de acceso multirregional), revise la configuración de bloqueo del acceso público para el punto de acceso multirregional.

> **Note**
>
> No puede editar la configuración de Block Public Access después de que se cree el punto de acceso multirregional. Por lo tanto, si va a bloquear el acceso público, asegúrese de que las aplicaciones funcionen correctamente sin acceso público antes de crear un punto de acceso multirregional.

Uso de una política de punto de acceso multirregional

La política de punto de acceso multirregional de ejemplo siguiente concede un acceso de usuario de IAM para mostrar y descargar archivos desde el punto de acceso multirregional. Para utilizar esta política de ejemplo, sustituya *`user input placeholders`* por su propia información.

```
{
  "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/
*"
         ]
      }
   ]
}
```

Para asociar la política de punto de acceso de varias regiones con el punto de acceso de varias regiones con la AWS Command Line Interface (AWS CLI), utilice el siguiente comando `put-multi-region-access-point-policy`. Para utilizar este comando de ejemplo, sustituya *`user input placeholders`* por su propia información. Cada punto de acceso de varias regiones solo puede tener una política, por lo que una solicitud realizada a la acción `put-multi-region-access-point-policy` sustituye cualquier política existente que esté asociada con el punto de acceso de varias regiones especificado.

AWS CLI

```
aws s3control put-multi-region-access-point-policy
--account-id 111122223333
--details { "Name": "amzn-s3-demo-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/*
\" ] } }" }
```

Para consultar los resultados de la operación anterior, utilice el siguiente comando:

AWS CLI

```
aws s3control describe-multi-region-access-point-operation
--account-id 111122223333
--request-token-arn requestArn
```

Para recuperar la política de punto de acceso de varias regiones, utilice el siguiente comando:

AWS CLI

```
aws s3control get-multi-region-access-point-policy
--account-id 111122223333
--name=amzn-s3-demo-bucket-MultiRegionAccessPoint
```

Editar la política del punto de acceso multirregional

La política de punto de acceso multirregional (escrita en JSON) proporciona acceso de almacenamiento a los buckets de Amazon S3 que se utilizan con este punto de acceso multirregional. Puede permitir o denegar que determinadas entidades principales realicen diversas acciones en el punto de acceso multirregional. Cuando una solicitud se enruta a un bucket a través del punto de acceso multirregional, se aplican las políticas de acceso para el punto de acceso multirregional y para los buckets. La política de acceso más restrictiva siempre tiene prioridad.

> **Note**
>
> Si un bucket contiene objetos que pertenecen a otras cuentas, la política de puntos de acceso de varias regiones no se aplica a los objetos que son propiedad de otras Cuentas de AWS.

Después de aplicar una política de punto de acceso multirregional, la política no se puede eliminar. Puede editar la política o crear una nueva política que sobrescriba la existente.

Para editar la política del punto de acceso multirregional

1. Inicie sesión AWS Management Console Management Console y abra la consola de Amazon S3 en https://console.aws.amazon.com/s3/.
2. En el panel de navegación izquierdo, elija Puntos de acceso de varias regiones.
3. Elija el nombre del punto de acceso de varias regiones para el que desea editar la política.
4. Elija la pestaña Permisos.
5. Desplácese hacia abajo hasta la sección Multi-Region Access Point policy (Política de punto de acceso multirregional). Elija Edit (Editar) para actualizar la política (en JSON).

6. Aparece la página Edit Multi-Region Access Point policy (Editar política de punto de acceso multirregional). Puede ingresar la política directamente en el campo de texto o puede elegir Add statement (Agregar instrucción) para seleccionar los elementos de la política de una lista desplegable.

> **Note**
>
> La consola muestra automáticamente el nombre de recurso de Amazon (ARN) del punto de acceso de varias regiones, que puede utilizar en la política. Para políticas de punto de acceso multirregional de ejemplo, consulte the section called “Ejemplos de política de punto de acceso multirregional”.

Ejemplos de política de punto de acceso multirregional

Los puntos de acceso de varias regiones de Amazon S3 admiten políticas de recursos de AWS Identity and Access Management (IAM). Puede usar estas políticas para controlar el uso del punto de acceso de varias regiones en función del recurso, del usuario o de otras condiciones. Para que una aplicación o un usuario puedan acceder a objetos a través de un punto de acceso multirregional, tanto el punto de acceso de varias regiones como el bucket subyacente deben permitir el mismo acceso.

Para permitir el mismo acceso al punto de acceso multirregional y al bucket subyacente, realice una de las siguientes acciones:

- (Recomendado) Para simplificar los controles de acceso al utilizar un punto de acceso multirregional de Amazon S3, delegue el control de acceso del bucket de Amazon S3 al punto de acceso multirregional. Para ver un ejemplo práctico, consulte el ejemplo 1 de esta sección.
- Agregue los mismos permisos contenidos en la política de puntos de acceso de varias regiones a la política del bucket subyacente.

> **Important**
>
> La delegación del control de acceso para un bucket a una política de punto de acceso multirregional no cambia el comportamiento del bucket cuando se accede al bucket a través del nombre de recurso de Amazon (ARN). Todas las operaciones realizadas directamente con respecto al bucket continuarán funcionando como antes. Las restricciones que se

incluyen en una política de punto de acceso de varias regiones solo se aplican a las solicitudes realizadas a través de ese punto de acceso multirregional.

Example 1: Delegar el acceso a puntos de acceso de varias regiones en la política de buckets (para la misma cuenta o entre cuentas)

La política de bucket de ejemplo siguiente permite acceso completo a puntos de acceso de varias regiones. Esto significa que todo el acceso a este bucket está controlado por las políticas asociadas a los puntos de acceso de varias regiones. Recomendamos configurar los buckets de esta manera para todos los casos de uso que no requieran acceso directo al bucket. Puede utilizar esta estructura de política de bucket para los puntos de acceso de varias regiones de la misma cuenta o de otra cuenta.

```
{
    "Version": "2012-10-17",
    "Statement" : [
    {
        "Effect": "Allow",
        "Principal" : { "AWS": "*" },
        "Action" : "*",
        "Resource" : [ "Bucket ARN", "Bucket ARN/*"],
        "Condition": {
            "StringEquals" : { "s3:DataAccessPointArn" : "MultiRegionAccessPoint_ARN" }
        }
    }]
}
```

Note

Si hay varios puntos de acceso de varias regiones a los que está concediendo acceso, asegúrese de mostrar cada punto de acceso de varias regiones.

Example 2: Conceder acceso a una cuenta a un punto de acceso de varias regiones en la política de punto de acceso de varias regiones

La política de punto de acceso multirregional siguiente concede a la cuenta *123456789012* permiso para enumerar y consultar los objetos contenidos en el punto de acceso multirregional definidos por el *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: Política de puntos de acceso multirregional que permite el listado de buckets

La siguiente política de punto de acceso multirregional siguiente concede el permiso *123456789012* de cuenta para enumerar los objetos contenidos en el punto de acceso multirregional definidos por el *MultiRegionAccessPoint_ARN*.

```
{
   "Version":"2012-10-17",
   "Statement": [
      {
       "Effect": "Allow",
       "Principal": {
        "AWS": "arn:aws:iam::123456789012:user/JohnDoe"
        },
        "Action": "s3:ListBucket",
        "Resource": "MultiRegionAccessPoint_ARN"
       }
    ]
}
```

Restricciones y limitaciones de puntos de acceso de varias regiones

Los puntos de acceso de varias regiones de Amazon S3 tienen en cuenta las siguientes restricciones y limitaciones:

- Nombres de puntos de acceso de varias regiones:
  - Deben ser únicos para cada cuenta de AWS
  - Debe comenzar con un número o una letra minúscula.
  - Deben tener entre 3 y 50 caracteres de longitud.
  - No puede comenzar ni terminar con un guion (`-`)
  - No puede contener guiones bajos (`_`), letras mayúsculas ni puntos (`.`)
  - No se pueden editar después de crearlos
- Amazon S3 genera alias de punto de acceso multirregional y no se pueden editar ni reutilizar.
- No se puede acceder a los datos a través de un punto de acceso multirregional mediante puntos de conexión de puerta de enlace. Sin embargo, no se puede acceder a los datos a través de un punto de acceso multirregional mediante puntos de conexión de interfaz. Para utilizar AWS PrivateLink, debe crear puntos de conexión de VPC. Para obtener más información, consulte Configuración de un punto de acceso de varias regiones para su uso con AWS PrivateLink.
- Para usar puntos de acceso de varias regiones con Amazon CloudFront, debe configurar el punto de acceso de varias regiones como un tipo de distribución `Custom Origin`. Para obtener más información sobre varios tipos de origen, consulte Uso de varios orígenes con distribuciones de CloudFront. Para obtener más información sobre el uso de los puntos de acceso de varias regiones con Amazon CloudFront, consulte Crear una aplicación activa-activa y basada en la proximidad en varias regiones en el Blog de almacenamiento de AWS.
- Requisitos mínimos del punto de acceso de varias regiones:
  - Transport Layer Security (TLS) v1.2
  - Versión de Signature 4 (SigV4A)

    Los puntos de acceso de varias regiones admiten la versión 4A de Signature. Esta versión SigV4 permite que las solicitudes se firmen para múltiples Regiones de AWS. Esta característica es útil en operaciones de API que podrían dar como resultado el acceso a datos desde una de varias regiones. Cuando utilice un SDK de AWS, proporcione las credenciales y las solicitudes a los puntos de acceso de varias regiones usarán la versión 4A de Signature sin configuración adicional. Asegúrese de comprobar la compatibilidad del SDK de AWS con el algoritmo SigV4a.

Para obtener más información acerca de SigV4A, consulte Cómo firmar solicitudes de API de AWS en la Referencia general de AWS.

> **Note**
>
> Para usar SigV4a con credenciales de seguridad temporales (por ejemplo, cuando se utilizan roles de AWS Identity and Access Management [IAM]), asegúrese de solicitar las credenciales temporales desde un punto de conexión regional en AWS Security Token Service (AWS STS). Si solicita credenciales temporales desde el punto de conexión AWS STS global (`sts.amazonaws.com`), primero debe configurar la compatibilidad de la región de los tokens de sesión para que el punto de conexión global sea válido en todas las Regiones de AWS. Para obtener más información, consulte Administración de AWS STS en una Región de AWS en la guía del usuario de IAM.

- Los puntos de acceso de varias regiones no admiten solicitudes realizadas de forma anónima.
- Limitaciones del punto de acceso de varias regiones:
  - No se admite IPv6.
  - No se admiten buckets de Amazon S3 en Outposts.
  - Los puntos de acceso de varias regiones admiten operaciones de copia con puntos de acceso de varias regiones solo como destino o cuando se utiliza el ARN del punto de acceso de varias regiones.
  - La característica de operaciones por lotes de S3 no se admite.
- Determinados SDK de AWS no se admiten. Para confirmar qué SDK de AWS son compatibles con los puntos de acceso de varias regiones, consulte Compatibilidad con los SDK de AWS.
- Service Quotas para puntos de acceso de varias regiones son de la siguiente manera:
  - Hay un máximo de 100 puntos de acceso de varias regiones por cuenta.
  - Hay un límite de 17 regiones para un único punto de acceso multirregional.
- Después de crearlo, no puede agregar, modificar ni eliminar buckets de la configuración del punto de acceso de varias regiones. Para cambiar los buckets, debe eliminar todo el punto de acceso de varias regiones y crear uno nuevo. Si se elimina un bucket entre cuentas en su punto de acceso de varias regiones, la única forma de volver a conectar ese bucket es volver a crear el bucket con el mismo nombre y región en esa cuenta.
- Los bucket subyacentes (en la misma cuenta) que se utilizan en un punto de acceso de varias regiones solo se pueden eliminar una vez eliminado un punto de acceso de varias regiones.

- Todas las solicitudes de plano de control para crear o mantener puntos de acceso de varias regiones se dirigen a la región `US West (Oregon)`. Para las solicitudes de plano de datos de puntos de acceso de varias regiones, no es necesario especificar las regiones.
- Para el plano de control de la conmutación por error del punto de acceso de varias regiones, las solicitudes deben dirigirse a una de las cinco regiones admitidas:
  - `US East (N. Virginia)`
  - `US West (Oregon)`
  - `Asia Pacific (Sydney)`
  - `Asia Pacific (Tokyo)`
  - `Europe (Ireland)`
- Su punto de acceso de varias regiones solo admite buckets en las siguientes Regiones de 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)`

Enrutamiento de solicitud de punto de acceso de varias regiones

Cuando realiza una solicitud a través de un punto de acceso de varias regiones, Amazon S3 determina cuál de los buckets asociados con el punto de acceso de varias regiones está más cerca. A continuación, Amazon S3 dirige la solicitud a ese bucket, independientemente de la AWS región en la que se encuentra.

Después de que el punto de acceso de varias regiones enrute la solicitud al bucket más cercano, Amazon S3 procesa la solicitud como si la hubiera realizado directamente hacia ese bucket. Los puntos de acceso de varias regiones no conocen el contenido de datos de un bucket de Amazon S3. Por lo tanto, es posible que el bucket que reciba la solicitud no contenga los datos solicitados. Para crear conjuntos de datos coherentes en los buckets de Amazon S3 asociados con un punto de acceso de varias regiones, puede configurar la replicación entre regiones (CRR) de S3. Entonces cualquier bucket puede cumplir con la solicitud correctamente.

Amazon S3 dirige las solicitudes de puntos de acceso de varias regiones de acuerdo con las siguientes reglas:

- Amazon S3 optimiza las solicitudes que se deben cumplir en función de la proximidad. Observa los buckets admitidos por el punto de acceso multirregional y retrasa la solicitud al bucket que tiene más cerca.
- Si la solicitud especifica un recurso existente (por ejemplo: `GetObject`), Amazon S3 no considera el nombre del objeto al cumplir la solicitud. Esto significa que incluso si existe un objeto puede en un bucket en el punto de acceso de varias regiones, su solicitud se podrá enrutar hacia un bucket que no contenga el objeto. Esto hará que el cliente reciba un mensaje de error 404.

  Para evitar errores 404, se recomienda configurar la replicación entre regiones (CRR) de S3 para los buckets. La replicación ayuda a resolver este potencial problema cuando el objeto que desea se encuentra en un bucket en el punto de acceso de varias regiones, pero no se encuentra en el bucket específico al que se dirigió la solicitud. Para obtener información acerca de la configuración de replicación, consulte Configuración de la replicación de bucket para utilizarla con puntos de acceso de varias regiones.

  Para asegurarse de que las solicitudes se satisfagan utilizando los objetos específicos que desea, también le recomendamos que active el control de versiones del bucket e incluya los ID de versión en sus solicitudes. Este enfoque ayuda a garantizar que dispone de la versión correcta del objeto que está buscando. Los buckets con el control de versiones habilitado también ayudan a recuperar

objetos que se hayan sobrescrito por error. Para obtener más información, consulte Usar el control de versiones S3 en buckets de S3.

- Si la solicitud es para crear un recurso (por ejemplo, `PutObject` o `CreateMultipartUpload`), Amazon S3 gestiona la solicitud con el bucket más cercano. Por ejemplo, piense en una empresa de vídeo que desea admitir cargas de vídeos de cualquier parte del mundo. Cuando un usuario crea una solicitud PUT al punto de acceso de varias regiones, el objeto se coloca en el bucket más próximo. Para que el vídeo cargado esté disponible para que otros usuarios de todo el mundo lo descarguen con la menor latencia, puede usar la CRR con replicación bidireccional. El uso de la CRR con replicación biregional mantiene sincronizado el contenido de todos los buckets asociados con el punto de acceso de varias regiones. Para obtener más información acerca de la replicación con puntos de acceso de varias regiones, consulte Configuración de la replicación de bucket para utilizarla con puntos de acceso de varias regiones.

Controles de conmutación por error de puntos de acceso de varias regiones de Amazon S3

Con los controles de conmutación por error de los puntos de acceso de varias regiones de Amazon S3, puede mantener la continuidad empresarial durante las interrupciones del tráfico regional y, al mismo tiempo, ofrecer a las aplicaciones una arquitectura multirregional para cumplir con las necesidades de cumplimiento y redundancia. Si el tráfico regional se interrumpe, puede utilizar los controles de conmutación por error de puntos de acceso de varias regiones para seleccionar qué Regiones de AWS detrás de un punto de acceso multirregional de Amazon S3 procesará las solicitudes de almacenamiento y acceso a los datos.

Para permitir la conmutación por error, puede configurar el punto de acceso multirregional en una configuración activa-pasiva, con tráfico fluyendo a la región activa en condiciones normales y una región pasiva en espera para la conmutación por error.

Por ejemplo, para realizar una conmutación por error en una Región de AWS de su elección, puede cambiar el tráfico de la región principal (activa) a la región secundaria (pasiva). En una configuración activo-pasiva como esta, un bucket está activo y acepta tráfico, mientras que el otro bucket es pasivo y no acepta tráfico. El bucket pasivo se utiliza para la recuperación de desastres. Al iniciar la conmutación por error, todo el tráfico (por ejemplo, solicitudes `GET` o `PUT`) se dirige al bucket en estado activo (en una región) y se aleja del bucket en estado pasivo (en otra región).

Si tiene habilitada la replicación entre regiones (CRR) de S3 con reglas de replicación bidireccional, puede mantener los buckets sincronizados durante una conmutación por error. Además, si tiene

habilitada la CRR en una configuración activa-activa, los puntos de acceso de varias regiones de Amazon S3 también pueden obtener datos de la ubicación del bucket más cercana, lo que mejora el rendimiento de la aplicación.

Compatibilidad con Región de AWS

Con los controles de conmutación por error de los puntos de acceso de varias regiones de Amazon S3, los buckets de S3 pueden estar en cualquiera de las 17 regiones en las que se admiten los puntos de acceso de varias regiones. Puede iniciar la conmutación por error en dos regiones a la vez.

> **Note**
>
> Aunque la conmutación por error solo se inicia entre solo dos regiones a la vez, puede actualizar de forma independiente los estados de enrutamiento de varias regiones al mismo tiempo en el punto de acceso multirregional.

Los temas siguientes demuestran cómo usar y administrar los controles de conmutación por error de los puntos de acceso de varias regiones de Amazon S3.

Temas

- Estado de enrutamiento de puntos de acceso de varias regiones de Amazon S3
- Uso de los controles de conmutación por error de punto de acceso de varias regiones de Amazon S3
- Errores de controles de conmutación por error de punto de acceso de varias regiones de Amazon S3

Estado de enrutamiento de puntos de acceso de varias regiones de Amazon S3

La configuración de la conmutación por error de los puntos de acceso de varias regiones de Amazon S3 determina el estado de enrutamiento de las Regiones de AWS que se utilizan con el punto de acceso de varias regiones. Puede configurar el punto de acceso multirregional de Amazon S3 para que esté en estado activo-activo o activo-pasivo.

- Activa-activa: en una configuración activa-activa, todas las solicitudes se envían automáticamente a la Región de AWS más cercana en el punto de acceso multirregional. Una vez que el punto

de acceso multirregional se haya configurado para estar en un estado activo-activo, todas las regiones pueden recibir tráfico. Si se produce una interrupción del tráfico en una configuración activa-activa, el tráfico de red se redirigirá automáticamente a una de las regiones activas.

- Activa-pasiva: en una configuración activa-pasiva, las regiones activas en el punto de acceso multirregional reciben tráfico y las pasivas no. Si tiene la intención de utilizar los controles de conmutación por error de S3 para iniciar la conmutación por error en una situación de desastre, configure los puntos de acceso de varias regiones en una configuración activa-pasiva mientras prueba y realiza la planificación de recuperación de desastres.

Uso de los controles de conmutación por error de punto de acceso de varias regiones de Amazon S3

En esta sección, se explica cómo administrar y utilizar los controles de la conmutación por error de puntos de acceso de varias regiones de Amazon S3 mediante la AWS Management Console.

Hay dos controles de conmutación por error en la sección Failover configuration (Configuración de conmutación por error) de la página de detalles del punto de acceso multirregional, en la AWS Management Console: Edit routing status (Editar estado de enrutamiento) y Failover (Conmutación por error). También puede utilizar estos controles del modo siguiente:

- Edit routing status (Editar estado de enrutamiento): puede editar manualmente los estados de enrutamiento de hasta 17 Regiones de AWS en una sola solicitud para el punto de acceso multirregional eligiendo Edit routing status (Editar estado de enrutamiento). Puede utilizar Edit routing status (Editar estado de enrutamiento) para los siguientes fines:
  - Para configurar o editar los estados de enrutamiento de una o más regiones en el punto de acceso multirregional
  - Para crear una configuración de conmutación por error para el punto de acceso multirregional mediante la configuración de dos regiones para que estén en estado activo-pasivo
  - Para realizar una conmutación por error manual en las regiones
  - Para cambiar manualmente el tráfico entre regiones
- Failover (Conmutación por error): cuando inicia la conmutación por error eligiendo Failover (Conmutación por error), solo actualiza los estados de enrutamiento de dos regiones que ya están configuradas para estar en un estado activo-pasivo. Durante una conmutación por error que haya iniciado al elegir Failover (Conmutación por error), los estados de enrutamiento entre las dos regiones se cambian automáticamente.

Edición del estado de enrutamiento de las regiones en el punto de acceso multirregional

Puede actualizar manualmente los estados de enrutamiento de hasta 17 Regiones de AWS en una sola solicitud para el punto de acceso multirregional, eligiendo Edit routing status (Editar estado de enrutamiento) en la sección Failover configuration (Configuración de conmutación por error) de la página de detalles del punto de acceso multirregional. Sin embargo, cuando inicia la conmutación por error eligiendo Failover (Conmutación por error), solo actualiza los estados de enrutamiento de dos regiones que ya están configuradas para estar en un estado activo-pasivo. Durante una conmutación por error que haya iniciado al elegir Failover (Conmutación por error), los estados de enrutamiento entre las dos regiones se cambian automáticamente.

Puede utilizar Edit routing status (Editar estado de enrutamiento) (tal como se describe en el procedimiento siguiente) para los siguientes fines:

- Para configurar o editar los estados de enrutamiento de una o más regiones en el punto de acceso multirregional
- Para crear una configuración de conmutación por error para el punto de acceso multirregional mediante la configuración de dos regiones para que estén en estado activo-pasivo
- Para realizar una conmutación por error manual en las regiones
- Para cambiar manualmente el tráfico entre regiones

Uso de la consola de S3

Para actualizar el estado de enrutamiento de las regiones en el punto de acceso multirregional

1. Inicie sesión en la consola de administración de AWS.
2. Abra la consola de Amazon S3 en https://console.aws.amazon.com/s3.
3. En el panel de navegación izquierdo, elija Puntos de acceso de varias regiones.
4. Elija el punto de acceso multirregional que desea actualizar.
5. Elija la pestaña Replication and failover (Replicación y conmutación por error).
6. Seleccione una o varias regiones de las que desee editar el estado de enrutamiento.

> **Note**
>
> Para iniciar la conmutación por error, al menos una Región de AWS debe estar designada como Active (Activa) y una región como Pasive (Pasiva) en el punto de acceso multirregional.

7. Elija Edit routing status (Editar estado de enrutamiento).
8. En el cuadro de diálogo que aparece, seleccione Active (Activo) o Passive (Pasivo) para ver el Routing status (Estado de enrutamiento) de cada región.

   Un estado activo permite que el tráfico se enrute a la región. Un estado pasivo impide que el tráfico se dirija a la región.

   Si crea una configuración de conmutación por error para el punto de acceso de varias regiones o inicia una conmutación por error, al menos una Región de AWS debe estar designada como Active (Activa) y una región como Pasive (Pasiva) en el punto de acceso multirregional.
9. Elija Save routing status (Guardar estado de enrutamiento). El tráfico tarda unos 2 minutos en redirigirse.

Después de enviar el estado de enrutamiento de la Regiones de AWS para el punto de acceso multirregional, puede verificar los cambios de estado de enrutamiento. Para verificar estos cambios, vaya a Amazon CloudWatch en https://console.aws.amazon.com/cloudwatch/ para monitorear el cambio del tráfico de solicitudes de datos de Amazon S3 (por ejemplo, solicitudes `GET` y `PUT`) entre las regiones activas y pasivas. Las conexiones existentes no se interrumpirán durante la conmutación por error. Las conexiones existentes continuarán hasta que alcancen el estado de éxito o error.

Utilización de la AWS CLI

> **Note**
>
> Puede ejecutar comandos de enrutamiento de la AWS CLI de puntos de acceso de varias regiones en cualquiera de estas cinco regiones:
>
> - `ap-southeast-2`
> - `ap-northeast-1`
> - `us-east-1`

- us-west-2
- eu-west-1

El comando de ejemplo siguiente actualiza la configuración actual de enrutamiento del punto de acceso multirregional. Para actualizar el estado activo o pasivo de un bucket, establezca el valor `TrafficDialPercentage` en `100` para activo y en `0` para pasivo. En este ejemplo, *DOC-EXAMPLE-BUCKET-1* se establece en activo y *DOC-EXAMPLE-BUCKET-2* en pasivo. Para utilizar este comando de ejemplo, sustituya *user input placeholders* por su propia información.

```
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
```

El comando de ejemplo siguiente obtiene la configuración actualizada de enrutamiento del punto de acceso multirregional. Para utilizar este comando de ejemplo, sustituya *user input placeholders* por su propia información.

```
aws s3control get-multi-region-access-point-routes
--region eu-west-1
--account-id 111122223333
--mrap MultiRegionAccessPoint_ARN
```

Inicio de la conmutación por error

Al iniciar la conmutación por error eligiendo Failover (Conmutación por error) en la sección de Failover configuration (Configuración de la conmutación por error) de la página de detalles del punto de acceso multirregional, el tráfico de solicitudes de Amazon S3 pasa automáticamente a una Región de AWS alternativa. El proceso de conmutación por error se completa en 2 minutos.

Puede iniciar una conmutación por error en dos Regiones de AWS a la vez (de las 17 regiones en las que se admiten puntos de acceso de varias regiones). A continuación, se registran los eventos de conmutación por error en AWS CloudTrail. Al finalizar la conmutación por error, puede monitorear el tráfico de Amazon S3 y cualquier actualización de enrutamiento de tráfico de la nueva región activa en Amazon CloudWatch.

> **Important**
>
> Para mantener todos los metadatos y objetos sincronizados en los buckets durante la replicación de datos, le recomendamos que cree reglas de replicación bidireccional y habilite la sincronización de modificaciones de réplicas antes de configurar los controles de conmutación por error.
> Las reglas de replicación bidireccional ayudan a garantizar que, cuando se escriben datos en el bucket de Amazon S3 al que se transfiere el tráfico por error, esos datos se repliquen de nuevo en el bucket de origen. La sincronización de modificaciones de réplicas ayuda a garantizar que los metadatos de los objetos también se sincronicen entre buckets durante la replicación bidireccional.
> Para obtener más información acerca de la configuración de replicación para admitir conmutación por error, consulte the section called “Replicación de buckets”.

Para iniciar la conmutación por error entre buckets replicados

1. Inicie sesión en la consola de administración de AWS.
2. Abra la consola de Amazon S3 en https://console.aws.amazon.com/s3.
3. En el panel de navegación izquierdo, elija Puntos de acceso de varias regiones.
4. Elija el punto de acceso de varias regiones que desea usar para iniciar la conmutación por error.
5. Elija la pestaña Replication and failover (Replicación y conmutación por error).
6. Desplácese hacia abajo hasta la sección Failover configuration (Configuración de conmutación por error) y seleccione dos Regiones de AWS.

   > **Note**
   >
   > Para iniciar la conmutación por error, al menos una Región de AWS debe estar designada como Active (Activa) y una región como Pasive (Pasiva) en el punto de acceso multirregional. Un estado activo permite que el tráfico se dirija a una región. Un estado pasivo impide que el tráfico se dirija a la región.

7. Elija Failover (Conmutación por error).
8. En el cuadro de diálogo, vuelva a elegir Failover (Conmutación por error) para iniciar el proceso de conmutación por error. Durante este proceso, los estados de enrutamiento de las dos regiones se cambian automáticamente. Todo el tráfico nuevo se dirige a la región que pasa a

ser activa y el tráfico deja de dirigirse a la región que pasa a ser pasiva. El tráfico tarda unos 2 minutos en redirigirse.

Tras iniciar el proceso de conmutación por error, puede verificar los cambios en el tráfico. Para verificar estos cambios, vaya a Amazon CloudWatch en https://console.aws.amazon.com/cloudwatch/ para monitorear el cambio del tráfico de solicitudes de datos de Amazon S3 (por ejemplo, solicitudes GET y PUT) entre las regiones activas y pasivas. Las conexiones existentes no se interrumpirán durante la conmutación por error. Las conexiones existentes continuarán hasta que alcancen el estado de éxito o error.

Cómo ver los controles de enrutamiento de puntos de acceso de varias regiones de Amazon S3

Uso de la consola de S3

Para ver los controles de enrutamiento de punto de acceso de varias regiones de Amazon S3

1. Inicie sesión en la consola de administración de AWS.
2. Abra la consola de Amazon S3 en https://console.aws.amazon.com/s3.
3. En el panel de navegación izquierdo, elija Puntos de acceso de varias regiones.
4. Elija el punto de acceso de varias regiones que desea revisar.
5. Elija la pestaña Replication and failover (Replicación y conmutación por error). En esta página se muestran los detalles y el resumen de la configuración de enrutamiento del punto de acceso multirregional, las reglas de replicación asociadas y las métricas de replicación. Puede ver el estado del enrutamiento de las regiones en la sección Failover configuration (Configuración de conmutación por error).

Utilización de la AWS CLI

El comando de la AWS CLI de ejemplo siguiente obtiene la configuración actual de la ruta del punto de acceso multirregional para la región especificada. Para utilizar este comando de ejemplo, sustituya *user input placeholders* por su propia información.

```
aws s3control get-multi-region-access-point-routes
--region eu-west-1
--account-id 111122223333
--mrap MultiRegionAccessPoint_ARN
```

> **Note**
>
> Este comando solo se puede ejecutar en estas cinco regiones:
>
> - `ap-southeast-2`
> - `ap-northeast-1`
> - `us-east-1`
> - `us-west-2`
> - `eu-west-1`

Errores de controles de conmutación por error de punto de acceso de varias regiones de Amazon S3

Al actualizar la configuración de conmutación por error del punto de acceso multirregional, es posible que se produzca uno de los siguientes errores:

- HTTP 400 Solicitud errónea: esto se puede producir si indica un ARN de punto de acceso de varias regiones no válido al actualizar la configuración de la conmutación por error. Puede confirmar el ARN de punto de acceso multirregional consultando la política de punto de acceso multirregional. Para revisar o actualizar la política de punto de acceso multirregional, consulte Edición de la política de punto de acceso de varias regiones. Este error también se puede producir si utiliza una cadena vacía o una cadena aleatoria al actualizar los controles de conmutación por error del punto de acceso multirregional de Amazon S3. Asegúrese de usar el formato de ARN de los puntos de acceso de varias regiones:

  `arn:aws:s3::`*`account-id`*`:accesspoint/`*`MultiRegionAccessPoint_alias`*

- HTTP 503 Slow Down: este error se produce si envía demasiadas solicitudes en un periodo corto de tiempo. Las solicitudes rechazadas producirán un error.
- HTTP 409 Conflict: este error se produce cuando dos o más solicitudes simultáneas de actualización de la configuración de enrutamiento se dirigen a un único punto de acceso multirregional. La primera solicitud se realiza correctamente, pero las demás solicitudes producen un error.
- HTTP 405 Method Not Allowed: este error se produce al seleccionar un punto de acceso multirregional con solo una Región de AWS al iniciar la conmutación por error. Debe seleccionar

dos regiones antes de poder iniciar la conmutación por error. En caso contrario, se devuelve un error.

Configuración de la replicación de bucket para utilizarla con puntos de acceso de varias regiones

Cuando realiza una solicitud a un punto de conexión del punto de acceso de varias regiones, Amazon S3 enruta automáticamente la solicitud hacia el bucket más cercano a usted. Amazon S3 no tiene en cuenta el contenido de la solicitud al tomar esta decisión. Si realiza una solicitud `GET` a un objeto, es posible que la solicitud se dirija a un bucket que no tiene una copia de este objeto. Si eso sucede, recibirá un error del código de estado HTTP 404 (No encontrado). Para obtener más información acerca del enrutamiento de solicitudes al punto de acceso de varias regiones, consulte the section called “Enrutamiento de solicitudes”.

Si desea que el punto de acceso multirregional pueda recuperar el objeto independientemente del bucket que recibe la solicitud, debe configurar la replicación entre regiones de Amazon S3 (CRR).

Por ejemplo, considere un punto de acceso multirregional con tres buckets:

- Un bucket denominado `my-bucket-usw2` en la región `us-west-2` que contiene el objeto `my-image.jpg`
- Un bucket denominado `my-bucket-aps1` en la región `ap-south-1` que contiene el objeto `my-image.jpg`
- Un bucket denominado `my-bucket-euc1` en la región `eu-central-1` que no contiene el objeto `my-image.jpg`

En esta situación, si realiza una `GetObject` solicitud para el objeto `my-image.jpg`, el éxito de esa solicitud depende de qué bucket reciba su solicitud. Dado que Amazon S3 no tiene en cuenta el contenido de la solicitud, es posible que enrute la solicitud `GetObject` al bucket `my-bucket-euc1` si ese bucket es el más cercano. Aunque el objeto se encuentre en un bucket en el punto de acceso multirregional, recibirá un error 404 Not Found (No encontrado) porque el bucket individual que recibió la solicitud no tenía el objeto.

La habilitación de la replicación entre regiones (CRR) ayuda a evitar este resultado. Con las reglas de replicación adecuadas, el objeto `my-image.jpg` se copia en el bucket `my-bucket-euc1`. Por lo tanto, si Amazon S3 enruta la solicitud a ese bucket, ahora puede recuperar el objeto.

La replicación funciona de forma normal con los buckets asignados a un punto de acceso de varias regiones. Amazon S3 no realiza ninguna manipulación de replicación especial con buckets que se encuentran en puntos de acceso de varias regiones. Para obtener más información acerca de la configuración de la replicación en los buckets, consulte Configuración de la replicación en directo.

Recomendaciones para utilizar la replicación con puntos de acceso de varias regiones

Para obtener el mejor rendimiento de replicación cuando se trabaja con puntos de acceso de varias regiones, se recomienda lo siguiente:

- Configure el control del tiempo de Replicación de S3 (S3 RTC). Para replicar los datos en distintas regiones dentro de un periodo predecible, puede utilizar S3 RTC. S3 RTC replica el 99,99 % de los objetos nuevos almacenados en Amazon S3) en un plazo de 15 minutos (con el respaldo de un acuerdo de nivel de servicio). Para obtener más información, consulte the section called “Uso del control de tiempo de replicación S3”. Hay cargos adicionales para S3 RTC. Para obtener información, consulte Precios de Amazon S3.
- Use la replicación bidireccional para permitir que los buckets sigan sincronizados cuando se actualicen a través del punto de acceso de varias regiones. Para obtener más información, consulte the section called “Crear una regla de replicación bidireccional para el punto de acceso de varias regiones”.
- Cree puntos de acceso de varias regiones entre cuentas para replicar los datos en buckets por separado Cuentas de AWS. Este enfoque proporciona una separación en el nivel de cuenta, de modo que se pueda acceder a los datos desde diferentes cuentas en diferentes regiones distintas del bucket de origen y replicarlos en ellas. La configuración de puntos de acceso de varias regiones entre cuentas no tiene costo adicional. Si es propietario de un bucket pero no es el propietario del punto de acceso de varias regiones, solo paga por los costos de transferencia y solicitud de datos. Los propietarios de puntos de acceso de varias regiones pagan los costos de enrutamiento de datos y aceleración de internet. Para obtener más información, consulte Precios de Amazon S3.
- Habilite la sincronización de modificaciones de réplicas para cada regla de replicación para mantener también sincronizados los cambios en los metadatos de los objetos. Para obtener más información, consulte Habilitación de sincronización de modificación de réplica.
- Habilite las métricas de Amazon CloudWatch para monitorizar los eventos de replicación. Se aplican tarifas de métricas de CloudWatch. Para obtener más información, consulte los precios de Amazon CloudWatch.

Temas

- Crear una regla de replicación unidireccional para el punto de acceso de varias regiones
- Crear una regla de replicación bidireccional para el punto de acceso de varias regiones
- Visualización de las reglas de replicación para el punto de acceso de varias regiones

Crear una regla de replicación unidireccional para el punto de acceso de varias regiones

Las reglas de replicación habilitan la copia automática y asincrónica de los objetos entre buckets. Una regla de replicación unidireccional ayuda a garantizar que los datos se repliquen por completo desde un bucket de origen de una Región de AWS hasta un bucket de destino de otra región. Cuando se configura la replicación unidireccional, se crea una regla de replicación desde el bucket de origen (DOC-EXAMPLE-BUCKET-1) al bucket de destino (DOC-EXAMPLE-BUCKET-2). Como todas las reglas de replicación, puede aplicar la regla de replicación unidireccional a todo el bucket de Amazon S3 o a un subconjunto de objetos filtrado por un prefijo o etiquetas de objeto.

> **Important**
>
> Recomendamos utilizar la replicación unidireccional si los usuarios solo van a consumir los objetos de los buckets de destino. Si sus usuarios van a cargar o modificar los objetos de sus buckets de destino, utilice la replicación bidireccional para mantener todos los buckets sincronizados. También se recomienda la replicación bidireccional si tiene previsto utilizar el punto de acceso de varias regiones para la conmutación por error. Para configurar la replicación bidireccional, consulte the section called “Crear una regla de replicación bidireccional para el punto de acceso de varias regiones”.

Para crear una regla de replicación unidireccional para el punto de acceso de varias regiones

1. Inicie sesión en la AWS Management Console y abra la consola de Amazon S3 en https://console.aws.amazon.com/s3/.
2. En el panel de navegación izquierdo, elija Puntos de acceso de varias regiones.
3. Elija el nombre del punto de acceso de varias regiones.
4. Elija la pestaña Replication and failover (Replicación y conmutación por error).
5. Desplácese a la sección Replication rules (Reglas de replicación) y, a continuación, elija Create replication rules (Crear reglas de replicación). Asegúrese de contar con los permisos suficientes para crear la regla de replicación; de lo contrario, se deshabilitará el control de versiones.

> **Note**
>
> Puede crear reglas de replicación solo para buckets de su cuenta. Para crear reglas de replicación para buckets externos, los propietarios de los buckets deben crear las reglas de replicación para dichos buckets.

6. En la página Crear reglas de replicación, elija la plantilla Replicar objetos de uno o más buckets de origen en uno o más buckets de destino.

> **Important**
>
> Al crear reglas de replicación mediante esta plantilla, sustituyen a las reglas de replicación existentes que ya estén asignadas al bucket.
> Para agregar o modificar las reglas de replicación existentes en lugar de sustituirlas, vaya a la pestaña Management (Administración) de cada bucket en la consola y, a continuación, edite las reglas en la sección Replication rules (Reglas de replicación). También puede agregar o modificar las reglas de replicación existentes mediante AWS CLI, los SDK o la API de REST. Para obtener más información, consulte Configuración de replicación.

7. En la sección Origen y destino, en Buckets de origen, seleccione uno o más buckets desde los que desee replicar objetos. Todos los buckets (de origen y destino) elegidos para la replicación deben tener habilitado el control de versiones de S3 y cada bucket debe residir en una Región de AWS diferente. Para obtener más información sobre el control de versiones de S3, consulte Uso de control de versiones en buckets de Amazon S3.

   En la sección Buckets de destino, seleccione uno o más buckets en los que desee replicar objetos.

> **Note**
>
> Asegúrese de contar con los permisos de lectura y replicación necesarios para establecer la replicación pues, de lo contrario, se producirán errores. Para obtener más información, consulte Creación de un rol de IAM.

8. En la sección Replication rule configuration (Configuración de reglas de replicación), elija si la regla de replicación estará Enabled (Habilitada) o Disabled (Desactivada) cuando se cree.

> **Note**
>
> No puede ingresar un nombre en el cuadro Replication rule name (Nombre de la regla de replicación). Los nombres de las reglas de replicación se generan en función de la configuración al crear la regla de replicación.

9. En la sección Scope (Alcance), elija el alcance adecuado para la replicación.

   - Para replicar todo el bucket, elija Apply to all objects in the bucket (Aplicar a todos los objetos del bucket).
   - Para replicar un subconjunto de objetos en el bucket, elija Limit the scope of this rule using one or more filters (Limitar el alcance de esta regla mediante uno o varios filtros).

     Puede filtrar los objetos utilizando un prefijo, etiquetas de objeto o una combinación de ambos.

     - Para limitar la replicación a todos los objetos que tienen nombres que empiezan con la misma cadena (por ejemplo, `pictures`), escriba un prefijo en el cuadro Prefix (Prefijo).

       Si utiliza un prefijo que es el nombre de una carpeta, debe introducir un delimitador, como `/` (barra inclinada) para indicar el nivel de la jerarquía (por ejemplo, `pictures/`). Para obtener más información acerca de los prefijos, consulte Organizar objetos con prefijos.
     - Para replicar todos los objetos que tienen una o varias etiquetas de objeto, elija Add tag (Agregar etiqueta) y escriba el par clave-valor en los cuadros. Para agregar otra etiqueta, repita el procedimiento. Para obtener más información acerca de las etiquetas de objeto, consulte Categorización del almacenamiento mediante etiquetas.

10. Desplácese hacia abajo hasta la sección Additional replication options (Opciones de replicación adicionales) y seleccione las opciones de replicación que desee aplicar.

> **Note**
>
> Le recomendamos que aplique las siguientes opciones:
>
> - Replication time control (RTC) (Control del tiempo de replicación): para replicar los datos en distintas regiones dentro de un periodo predecible, puede utilizar el control del tiempo de Replicación de S3 (S3 RTC). S3 RTC replica el 99,99 % de los objetos nuevos almacenados en Amazon S3) en un plazo de 15 minutos (con el respaldo de un acuerdo de nivel de servicio). Para obtener más información, consulte the section called “Uso del control de tiempo de replicación S3”.

- Replication metrics and notifications (Notificaciones y métricas de replicación): habilite las métricas de Amazon CloudWatch para monitorear los eventos de replicación.
- Replicación de marcador de eliminación: se replicarán los marcadores de eliminación creados por las operaciones de eliminación de S3. Los marcadores de eliminación creados por las reglas del ciclo de vida no se replican. Para obtener más información, consulte Replicación de marcadores de eliminación entre buckets.

Se aplican cargos adicionales por las métricas y las notificaciones de Replicación de S3 RTC y CloudWatch. Para obtener más información, consulte Precios de Amazon S3 y Precios de Amazon CloudWatch.

11. Si está escribiendo una nueva regla de replicación que sustituya a una existente, seleccione I acknowledge that by choosing Create replication rules, these existing replication rules will be overwritten (Reconozco que, al elegir Crear reglas de replicación, estas reglas de replicación existentes se sobrescribirán).
12. Elija Crear reglas de replicación para crear y guardar la nueva regla de replicación unidireccional.

Crear una regla de replicación bidireccional para el punto de acceso de varias regiones

Las reglas de replicación habilitan la copia automática y asincrónica de los objetos entre buckets. Una regla de replicación bidireccional garantiza que los datos se sincronicen por completo entre dos o más buckets en diferentes Regiones de AWS. Cuando se configura la replicación bidireccional, se crea una regla de replicación desde el bucket de origen (DOC-EXAMPLE-BUCKET-1) al bucket que contiene las réplicas (DOC-EXAMPLE-BUCKET-2). Luego, se crea una segunda regla de replicación desde el bucket que contiene las réplicas (DOC-EXAMPLE-BUCKET-2) al bucket de origen (DOC-EXAMPLE-BUCKET-1).

Como todas las reglas de replicación, puede aplicar la regla de replicación bidireccional a todo el bucket de Amazon S3 o a un subconjunto de objetos filtrado por un prefijo o etiquetas de objeto. Puede mantener también sincronizados los cambios en los metadatos de los objetos mediante la habilitación de la sincronización de modificaciones de réplicas para cada regla de replicación. Puede habilitar la sincronización de modificaciones de réplicas a través de la consola de Amazon S3, la AWS CLI, los SDK de AWS, la API de REST de Amazon S3 o AWS CloudFormation.

Para monitorear el progreso de la replicación de los objetos y los metadatos de los objetos en Amazon CloudWatch, habilite las métricas y notificaciones de Replicación de S3. Para obtener más información, consulte Monitoreo del progreso con métricas de replicación y notificaciones de eventos de Amazon S3.

Para crear una regla de replicación bidireccional para el punto de acceso multirregional

1. Inicie sesión en la AWS Management Console y abra la consola de Amazon S3 en https://console.aws.amazon.com/s3/.
2. En el panel de navegación izquierdo, elija Puntos de acceso de varias regiones.
3. Elija el nombre del punto de acceso de varias regiones que desea actualizar.
4. Elija la pestaña Replication and failover (Replicación y conmutación por error).
5. Desplácese a la sección Replication rules (Reglas de replicación) y, a continuación, elija Create replication rules (Crear reglas de replicación).
6. En la página Create replication rules (Crear reglas de replicación), elija la plantilla Replicate objects among all specified buckets (Replicar objetos entre todos los buckets especificados). La plantilla Replicate objects among all specified buckets (Replicar objetos entre todos los buckets especificados) configura la replicación bidireccional (con capacidades de conmutación por error) para los buckets.

   > **Important**
   >
   > Al crear reglas de replicación mediante esta plantilla, sustituyen a las reglas de replicación existentes que ya estén asignadas al bucket.
   > Para agregar o modificar las reglas de replicación existentes en lugar de sustituirlas, vaya a la pestaña Management (Administración) de cada bucket en la consola y, a continuación, edite las reglas en la sección Replication rules (Reglas de replicación). También puede agregar o modificar las reglas de replicación existentes mediante la AWS CLI, los SDK de AWS o la API de REST de Amazon S3. Para obtener más información, consulte Configuración de replicación.

7. En la sección Buckets, seleccione al menos dos buckets desde los que desee replicar objetos. Todos los buckets elegidos para la replicación deben tener habilitado el control de versiones de S3 y cada bucket debe residir en una Región de AWS diferente. Para obtener más información sobre el control de versiones de S3, consulte Uso de control de versiones en buckets de Amazon S3.

> **Note**
>
> Asegúrese de contar con los permisos de lectura y replicación necesarios para establecer la replicación pues, de lo contrario, se producirán errores. Para obtener más información, consulte Creación de un rol de IAM.

8. En la sección Replication rule configuration (Configuración de reglas de replicación), elija si la regla de replicación estará Enabled (Habilitada) o Disabled (Desactivada) cuando se cree.

> **Note**
>
> No puede ingresar un nombre en el cuadro Replication rule name (Nombre de la regla de replicación). Los nombres de las reglas de replicación se generan en función de la configuración al crear la regla de replicación.

9. En la sección Scope (Alcance), elija el alcance adecuado para la replicación.

   - Para replicar todo el bucket, elija Apply to all objects in the bucket (Aplicar a todos los objetos del bucket).
   - Para replicar un subconjunto de objetos en el bucket, elija Limit the scope of this rule using one or more filters (Limitar el alcance de esta regla mediante uno o varios filtros).

     Puede filtrar los objetos utilizando un prefijo, etiquetas de objeto o una combinación de ambos.

     - Para limitar la replicación a todos los objetos que tienen nombres que empiezan con la misma cadena (por ejemplo, `pictures`), escriba un prefijo en el cuadro Prefix (Prefijo).

       Si utiliza un prefijo que es el nombre de una carpeta, debe usar una `/` (barra inclinada) como último carácter (por ejemplo, `pictures/`).
     - Para replicar todos los objetos que tienen una o varias etiquetas de objeto, elija Add tag (Agregar etiqueta) y escriba el par clave-valor en los cuadros. Para agregar otra etiqueta, repita el procedimiento. Para obtener más información acerca de las etiquetas de objeto, consulte Categorización del almacenamiento mediante etiquetas.

10. Desplácese hacia abajo hasta la sección Additional replication options (Opciones de replicación adicionales) y seleccione las opciones de replicación que desee aplicar.

> **Note**
>
> Le recomendamos que aplique las siguientes opciones, especialmente si tiene la intención de configurar el punto de acceso multirregional para admitir la conmutación por error:
>
> - Replication time control (RTC) (Control del tiempo de replicación): para replicar los datos en distintas regiones dentro de un periodo predecible, puede utilizar el control del tiempo de Replicación de S3 (S3 RTC). S3 RTC replica el 99,99 % de los objetos nuevos almacenados en Amazon S3) en un plazo de 15 minutos (con el respaldo de un acuerdo de nivel de servicio). Para obtener más información, consulte the section called “Uso del control de tiempo de replicación S3”.
> - Replication metrics and notifications (Notificaciones y métricas de replicación): habilite las métricas de Amazon CloudWatch para monitorear los eventos de replicación.
> - Replicación de marcador de eliminación: se replicarán los marcadores de eliminación creados por las operaciones de eliminación de S3. Los marcadores de eliminación creados por las reglas del ciclo de vida no se replican. Para obtener más información, consulte Replicación de marcadores de eliminación entre buckets.
> - Replica modification sync (Sincronización de modificación de réplicas): habilite la sincronización de modificaciones de réplicas para cada regla de replicación para mantener también sincronizados los cambios en los metadatos de los objetos. Para obtener más información, consulte Habilitación de sincronización de modificación de réplica.
>
> Se aplican cargos adicionales por las métricas y las notificaciones de Replicación de S3 RTC y CloudWatch. Para obtener más información, consulte Precios de Amazon S3 y Precios de Amazon CloudWatch.

11. Si está escribiendo una nueva regla de replicación que sustituya a una existente, seleccione I acknowledge that by choosing Create replication rules, these existing replication rules will be overwritten (Reconozco que, al elegir Crear reglas de replicación, estas reglas de replicación existentes se sobrescribirán).
12. Elija Create replication rules (Crear reglas de replicación) para crear y guardar las nuevas reglas de replicación bidireccional.

Visualización de las reglas de replicación para el punto de acceso de varias regiones

Con los puntos de acceso de varias regiones, puede configurar reglas de replicación unidireccionales o bidireccionales. Para obtener información sobre cómo administrar las reglas de replicación, consulte Administración de reglas de replicación mediante la consola de Amazon S3.

Para ver las reglas de replicación para el punto de acceso de varias regiones

1. Inicie sesión en la AWS Management Console y abra la consola de Amazon S3 en https://console.aws.amazon.com/s3/.
2. En el panel de navegación izquierdo, elija Puntos de acceso de varias regiones.
3. Elija el nombre del punto de acceso de varias regiones.
4. Elija la pestaña Replication and failover (Replicación y conmutación por error).
5. Desplácese hacia abajo hasta la sección Reglas de replicación. En esta sección se enumeran todas las reglas de replicación que se han creado para el punto de acceso de varias regiones.

> **Note**
>
> Si ha agregado un bucket de otra cuenta a este punto de acceso de varias regiones, debe contar con el permiso `s3:GetBucketReplication` del propietario del bucket para ver las reglas de replicación de ese bucket.

Uso de puntos de acceso de varias regiones con operaciones API admitidas

Amazon S3 proporciona un conjunto de operaciones para administrar puntos de acceso de varias regiones. Amazon S3 procesa algunas de estas operaciones de forma sincrónica y otras asíncrona. Cuando se invoca una operación asincrónica, Amazon S3 primero autoriza sincrónicamente la operación solicitada. Si la autorización se realiza correctamente, Amazon S3 devuelve un token que puede utilizar para realizar un seguimiento del progreso y los resultados de la operación solicitada.

> **Note**
>
> Las solicitudes que se realizan a través de la consola de Amazon S3 siempre son sincrónicas. La consola espera hasta que se complete la solicitud antes de permitirle enviar otra solicitud.

Puede ver el estado actual y los resultados de las operaciones asíncronas mediante la consola, o puede usar `DescribeMultiRegionAccessPointOperation` en la AWS CLI, los SDK de AWS o la API de REST. Amazon S3 proporciona un token de seguimiento en la respuesta a una operación asincrónica. Incluya ese token de seguimiento como argumento para `DescribeMultiRegionAccessPointOperation`. Si incluye el token de seguimiento, Amazon S3 devuelve el estado actual y los resultados de la operación especificada, incluidos los errores o la información pertinente de los recursos. Amazon S3 realiza operaciones de `DescribeMultiRegionAccessPointOperation` de forma sincrónica.

Todas las solicitudes de plano de control para crear o mantener puntos de acceso de varias regiones se dirigen a la región `US West (Oregon)`. Para las solicitudes de plano de datos de puntos de acceso de varias regiones, no es necesario especificar las regiones. Para el plano de control de la conmutación por error del punto de acceso de varias regiones, la solicitud debe dirigirse a una de las cinco regiones admitidas. Para obtener más información acerca de la regiones admitidas por el punto de acceso multiregional, consulte Restricciones y limitaciones de puntos de acceso de varias regiones.

Además, deberá otorgar el permiso `s3:ListAllMyBuckets` al usuario, rol u otra entidad de AWS Identity and Access Management (IAM) que realice una solicitud para administrar un punto de acceso de varias regiones.

Los ejemplos siguientes muestran cómo utilizar puntos de acceso de varias regiones con operaciones compatibles en Amazon S3.

Temas

- Compatibilidad de punto de acceso de varias regiones con Servicios de AWS y los SDK de AWS
- Compatibilidad de punto de acceso de varias regiones con operaciones de S3
- Consultar la configuración de enrutamiento de puntos de acceso de varias regiones
- Actualizar la política de bucket de Amazon S3 subyacente
- Actualizar una configuración de enrutamiento del punto de acceso multirregional

- Añadir un objeto a un bucket en el punto de acceso de varias regiones
- Recuperar objetos de un punto de acceso de varias regiones
- Enumerar los objetos que están almacenados en un bucket subyacente a su punto de acceso de varias regiones
- Utilice una URL prefirmada con puntos de acceso de varias regiones
- Uso de un bucket que se configura con pago por solicitante con puntos de acceso de varias regiones

Compatibilidad de punto de acceso de varias regiones con Servicios de AWS y los SDK de AWS

Para usar un punto de acceso de varias regiones con aplicaciones que requieren un nombre de bucket de Amazon S3, utilice el nombre de recurso de Amazon (ARN) del punto de acceso de varias regiones al realizar solicitudes con un SDK de AWS. Para comprobar si los SDK de AWS son compatibles con los puntos de acceso de varias regiones, consulte Compatibilidad con los SDK de AWS.

Compatibilidad de punto de acceso de varias regiones con operaciones de S3

Puede utilizar las siguientes operaciones de la API del plano de datos de Amazon S3 para realizar acciones en los objetos de los buckets asociados a su punto de acceso de varias regiones. Las siguientes operaciones de S3 pueden aceptar ARN de punto de acceso multirregional:

- AbortMultipartUpload
- CompleteMultipartUpload
- CreateMultipartUpload
- DeleteObject
- DeleteObjectTagging
- GetObject
- GetObjectAcl
- GetObjectLegalHold
- GetObjectRetention
- GetObjectTagging

- HeadObject
- ListMultipartUploads
- ListObjectsV2
- ListParts
- PutObject
- PutObjectAcl
- PutObjectLegalHold
- PutObjectRetention
- PutObjectTagging
- RestoreObject
- UploadPart

> **Note**
>
> Los puntos de acceso de varias regiones admiten operaciones de copia con puntos de acceso de varias regiones solo como destino o cuando se utiliza el ARN del punto de acceso de varias regiones.

Puede utilizar las siguientes operaciones del plano de control de Amazon S3 para crear y administrar sus puntos de acceso de varias regiones:

- CreateMultiRegionAccessPoint
- DescribeMultiRegionAccessPointOperation
- GetMultiRegionAccessPoint
- GetMultiRegionAccessPointPolicy
- GetMultiRegionAccessPointPolicyStatus
- GetMultiRegionAccessPointRoutes
- ListMultiRegionAccessPoints
- PutMultiRegionAccessPointPolicy
- SubmitMultiRegionAccessPointRoutes

Consultar la configuración de enrutamiento de puntos de acceso de varias regiones

AWS CLI

El comando de ejemplo siguiente recupera la configuración de enrutamiento del punto de acceso multirregional para que pueda ver los estados de enrutamiento actuales de los buckets. Para utilizar este comando de ejemplo, sustituya *user input placeholders* por su propia información.

```
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

El código de SDK para Java siguiente recupera la configuración de enrutamiento del punto de acceso multirregional para que pueda ver los estados de enrutamiento actuales de los buckets. Para utilizar esta sintaxis de ejemplo, sustituya *user input placeholders* por su propia información.

```
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

El código de SDK para JavaScript siguiente recupera la configuración de enrutamiento del punto de acceso multirregional para que pueda ver los estados de enrutamiento actuales de los buckets. Para utilizar esta sintaxis de ejemplo, sustituya *user input placeholders* por su propia información.

```
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

El código de SDK para Python siguiente recupera la configuración de enrutamiento del punto de acceso multirregional para que pueda ver los estados de enrutamiento actuales de los buckets. Para utilizar esta sintaxis de ejemplo, sustituya *user input placeholders* por su propia información.

```
s3.get_multi_region_access_point_routes(
        AccountId=111122223333,
        Mrap=arn:aws:s3::111122223333:accesspoint/abcdef0123456.mrap)['Routes']
```

Actualizar la política de bucket de Amazon S3 subyacente

Para conceder el acceso adecuado, también debe actualizar la política de bucket de Amazon S3 subyacente. En el siguiente ejemplo se delega el control de acceso a la política de punto de acceso de varias regiones. Tras delegar el control de acceso a la política de punto de acceso de varias

regiones, la política de bucket ya no se utiliza para el control de acceso cuando las solicitudes se realizan a través de ese punto de acceso de varias regiones.

A continuación, se muestra un ejemplo de política de bucket que delega el control de acceso a la política de punto de acceso multirregional. Para utilizar esta política de bucket de ejemplo, sustituya *`user input placeholders`* por su propia información. Para aplicar esta política mediante el comando AWS CLI `put-bucket-policy`, tal como se muestra en el siguiente ejemplo, guarde la política en un archivo, por ejemplo, `policy.json`.

```
{
  "Version": "2012-10-17",
  "Statement": {
    "Principal": { "AWS": "*" },
    "Effect": "Allow",
    "Action": ["s3:*"],
    "Resource": ["arn:aws:s3:::111122223333/*", "arn:aws:s3:::amzn-s3-demo-bucket"],
    "Condition": {
      "StringEquals": {
        "s3:DataAccessPointAccount": "444455556666"
      }
    }
  }
}
```

El comando de ejemplo `put-bucket-policy` siguiente asocia la política de bucket de S3 actualizada al bucket de S3:

```
aws s3api put-bucket-policy
  --bucket amzn-s3-demo-bucket
  --policy file:///tmp/policy.json
```

Actualizar una configuración de enrutamiento del punto de acceso multirregional

El comando de ejemplo siguiente actualiza la configuración de enrutamiento del punto de acceso multirregional. Puede ejecutar comandos de enrutamiento de puntos de acceso de varias regiones en cualquiera de las siguientes cinco regiones:

- `ap-southeast-2`
- `ap-northeast-1`
- `us-east-1`

- us-west-2
- eu-west-1

En una configuración de enrutamiento de puntos de acceso de varias regiones, puede configurar los buckets en un estado de enrutamiento activo o pasivo. Los buckets activos reciben tráfico, mientras que los buckets pasivos no. Puede establecer el estado de enrutamiento de un bucket configurando el valor `TrafficDialPercentage` del bucket en `100` para activo o `0` para pasivo.

AWS CLI

El comando de ejemplo siguiente actualiza la configuración de enrutamiento del punto de acceso multirregional. En este ejemplo, *amzn-s3-demo-bucket1* se establece en estado activo y *amzn-s3-demo-bucket2* en pasivo. Para utilizar este comando de ejemplo, sustituya *user input placeholders* por su propia información.

```
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=amzn-s3-demo-bucket1,TrafficDialPercentage=100
                Bucket=amzn-s3-demo-bucket2,TrafficDialPercentage=0
```

SDK for Java

El código de SDK para Java siguiente actualiza la configuración de enrutamiento del punto de acceso multirregional. Para utilizar esta sintaxis de ejemplo, sustituya *user input placeholders* por su propia información.

```
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

El código de SDK para JavaScript siguiente actualiza la configuración de enrutamiento del punto de acceso multirregional. Para utilizar esta sintaxis de ejemplo, sustituya *user input placeholders* por su propia información.

```
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: 'amzn-s3-demo-bucket1',
            TrafficDialPercentage: 0,
          },
        ],
      })
    )
```

```
    console.log('Success', data)
    return data
  } catch (err) {
    console.log('Error', err)
  }
}

run()
```

SDK for Python

El código de SDK para Python siguiente actualiza la configuración de enrutamiento del punto de acceso multirregional. Para utilizar esta sintaxis de ejemplo, sustituya *`user input placeholders`* por su propia información.

```
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
        }])
```

Añadir un objeto a un bucket en el punto de acceso de varias regiones

Para añadir un objeto al bucket asociado al punto de acceso de varias regiones, puede utilizar la operación PutObject. Para mantener sincronizados todos los buckets del punto de acceso de varias regiones, habilite la replicación entre regiones.

> **Note**
>
> Para utilizar esta operación, debe tener el permiso `s3:PutObject` para el punto de acceso de varias regiones. Para obtener más información acerca de los requisitos de los permisos del punto de acceso de varias regiones, consulte Permisos.

AWS CLI

El siguiente ejemplo de solicitud de plano de datos carga *example.txt* en el punto de acceso de varias regiones especificado. Para utilizar este ejemplo, reemplace los *user input placeholders* con su propia información.

```
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!'
)
```

Recuperar objetos de un punto de acceso de varias regiones

Para recuperar objetos del punto de acceso de varias regiones, puede utilizar la operaciónGetObject.

> **Note**
>
> Para utilizar esta operación de la API, debe tener el permiso `s3:GetObject` para el punto de acceso de varias regiones. Para obtener más información acerca de los requisitos de los permisos del punto de acceso de varias regiones, consulte Permisos.

AWS CLI

El siguiente ejemplo de solicitud de plano de datos recupera *example.txt* del punto de acceso de varias regiones especificado y lo descarga como *downloaded_example.txt*. Para utilizar este ejemplo, reemplace los *user input placeholders* con su propia información.

```
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'
)
```

Enumerar los objetos que están almacenados en un bucket subyacente a su punto de acceso de varias regiones

Para devolver una lista de objetos que están almacenados en un bucket subyacente a su punto de acceso de varias regiones, use la operación ListObjectsV2. En el siguiente comando de ejemplo, todos los objetos del punto de acceso de varias regiones especificado se enumeran con el ARN del punto de acceso de varias regiones. En este caso, el ARN del punto de acceso de varias regiones es:

```
arn:aws:s3::123456789012:accesspoint/abcdef0123456.mrap
```

> **Note**
>
> Para utilizar esta operación de la API, debe tener el permiso `s3:ListBucket` para el punto de acceso de varias regiones y el bucket subyacente. Para obtener más información acerca de los requisitos de los permisos del punto de acceso de varias regiones, consulte Permisos.

AWS CLI

En el siguiente ejemplo de solicitud de plano de datos se enumeran los objetos del bucket que subyace al punto de acceso de varias regiones especificado por el ARN. Para utilizar este ejemplo, reemplace los *user input placeholders* con su propia información.

```
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'
)
```

Utilice una URL prefirmada con puntos de acceso de varias regiones

Puede usar una URL prefirmada para generar una URL que permita a los demás acceder a los buckets de Amazon S3 a través de un punto de acceso de varias regiones de Amazon S3. Cuando crea una URL prefirmada, la asocia a una acción de objeto específica, como una carga de S3 (`PutObject`) o una descarga de S3 (`GetObject`). Puede compartir la URL prefirmada y cualquiera que tenga acceso a ella puede realizar la acción incrustada en la URL como si fuera el usuario de la firma original.

Las URL prefirmadas tienen fecha de vencimiento. Cuando se alcance el tiempo del vencimiento, la URL dejará de funcionar.

Antes de utilizar los puntos de acceso de varias regiones de S3 con URL prefirmadas, compruebe la Compatibilidad del SDK de AWS con el algoritmo SigV4A. Compruebe que la versión del SDK sea compatible con SigV4A como implementación de firma que se utiliza para firmar las solicitudes de Región de AWS globales. Para obtener más información sobre el uso de las URL prefirmadas con Amazon S3, consulte Compartir objetos mediante URL prefirmadas.

En los siguientes ejemplos, se muestra cómo puede utilizar puntos de acceso de varias regiones con URL prefirmadas. Para utilizar estos ejemplos, sustituya *`user input placeholders`* por su propia información.

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**
>
> Para usar SigV4a con credenciales de seguridad temporales (por ejemplo, cuando se utilizan roles de IAM), asegúrese de solicitar las credenciales temporales desde un punto de conexión regional en AWS Security Token Service (AWS STS), en lugar de un punto de conexión global. Si usa el punto de conexión global para AWS STS (`sts.amazonaws.com`), AWS STS generará credenciales temporales a partir de un punto de conexión global, lo cual no es compatible con Sig4A. Como resultado, obtendrá un error. Para resolver este problema, utilice cualquiera de los Puntos de conexión regionales para AWS STS.

Uso de un bucket que se configura con pago por solicitante con puntos de acceso de varias regiones

Si el bucket de S3 asociado a sus puntos de acceso de varias regiones está configurado para usar Pago por solicitante, el solicitante pagará tanto por la solicitud de bucket y la descarga, como por los costos relacionados con los puntos de acceso de varias regiones. Para obtener más información, consulte Precios de Amazon S3.

Este es un ejemplo de una solicitud de plano de datos a un punto de acceso de varias regiones que está conectado a un bucket Pago por solicitante.

AWS CLI

Para descargar objetos de un punto de acceso de varias regiones que está conectado a un bucket Pago por solicitante, especifique `--request-payer requester` como parte de la solicitud get-object. También debe especificar el nombre del archivo en el bucket, así como la ubicación en la que debe almacenarse.

```
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

Para descargar objetos de un punto de acceso de varias regiones que está conectado a un bucket Pago por solicitante, especifique `RequestPayer.REQUESTER` como parte de la solicitud `GetObject`. También debe especificar el nombre del archivo en el bucket, así como la ubicación en la que debe almacenarse.

```
GetObjectResponse getObjectResponse = s3Client.getObject(GetObjectRequest.builder()
    .key("example-file.txt")
    .bucket("arn:aws:s3::
123456789012:accesspoint/abcdef0123456.mrap")
    .requestPayer(RequestPayer.REQUESTER)
    .build()
).response();
```

Monitoreo y registro de solicitudes realizadas a través de un punto de acceso de varias regiones a los recursos subyacentes

Amazon S3 registra las solicitudes realizadas a través de los puntos de acceso de varias regiones y las solicitudes realizadas a las operaciones de la API que los administran, tales como `CreateMultiRegionAccessPoint` y `GetMultiRegionAccessPointPolicy`. Las solicitudes realizadas a Amazon S3 través de un punto de acceso de varias regiones aparecen en los registros de acceso de servidor de Amazon S3 y en los registros de AWS CloudTrail con el nombre de host del punto de acceso de varias regiones. El nombre de host de un punto de acceso toma la forma *`MRAP_alias`*`.accesspoint.s3-global.amazonaws.com`. Por ejemplo, supongamos que tiene la siguiente configuración de bucket y punto de acceso de varias regiones:

- Un bucket denominado `my-bucket-usw2` en la región `us-west-2` que contiene el objeto `my-image.jpg`.
- Un bucket denominado `my-bucket-aps1` en la región `ap-south-1` que contiene el objeto `my-image.jpg`.
- Un bucket denominado `my-bucket-euc1` en la región `eu-central-1` que no contiene un objeto denominado `my-image.jpg`.
- Un punto de acceso de varias regiones denominado `my-mrap` con el alias `mfzwi23gnjvgw.mrap` que está configurado para satisfacer las solicitudes de los tres buckets.
- El ID de la cuenta de AWS es `123456789012`.

Una solicitud realizada para recuperar `my-image.jpg` directamente a través de los buckets aparece en los registros con el nombre de host *`bucket_name`*`.s3.`*`Region`*`.amazonaws.com`.

Si realiza la solicitud a través del punto de acceso de varias regiones, Amazon S3 determina primero cuál de los buckets de las diferentes regiones está más cerca. Una vez que Amazon S3 determina qué bucket utilizar para gestionar la solicitud, envía la solicitud a ese bucket y registra la

operación utilizando el nombre de host del punto de acceso de varias regiones. En este ejemplo, si Amazon S3 retransmitió la solicitud a `my-bucket-aps1`, sus registros reflejarían una solicitud `GET` correcta para `my-image.jpg` desde `my-bucket-aps1`, usando un nombre de host de `mfzwi23gnjvgw.mrap.accesspoint.s3-global.amazonaws.com`.

> **Important**
>
> Los puntos de acceso de varias regiones no conocen el contenido de datos de los buckets subyacentes. Por lo tanto, es posible que el bucket que reciba la solicitud no contenga los datos solicitados. Por ejemplo, si Amazon S3 determina que el bucket de `my-bucket-euc1` está más cerca, los registros reflejarían una solicitud `GET` errónea para `my-image.jpg` desde `my-bucket-euc1`, usando un nombre de host de `mfzwi23gnjvgw.mrap.accesspoint.s3-global.amazonaws.com`. Si, en cambio, la solicitud se dirigió hacia `my-bucket-usw2`, sus registros indicarían un solicitud `GET` correcta.

Para obtener más información acerca de los registros de acceso al servidor de Amazon S3, consulte Registro de solicitudes con registro de acceso al servidor. Para obtener más información sobre AWS CloudTrail, consulte ¿Qué es AWS CloudTrail? en la Guía del usuario de AWS CloudTrail.

Monitoreo y registro de solicitudes realizadas a las operaciones de la API de administración de puntos de acceso de varias regiones

Amazon S3 ofrece varias operaciones de la API para administrar puntos de acceso de varias regiones, como `CreateMultiRegionAccessPoint` y `GetMultiRegionAccessPointPolicy`. Cuando realiza estas solicitudes a estas operaciones de la API utilizando la AWS Command Line Interface (AWS CLI), los SDK de AWS o la API de REST de Amazon S3, Amazon S3 procesa estas solicitudes de forma asincrónica. Si tiene los permisos adecuados para la solicitud, Amazon S3 devuelve un token para estas solicitudes. Puede usar este token con `DescribeAsyncOperation` para ayudarle a ver el estado de las operaciones asincrónicas en curso. Amazon S3 procesa las solicitudes de `DescribeAsyncOperation` de forma sincrónica. Puede utilizar la consola, AWS CLI, los SDK o la API de REST de Amazon S3 para ver el estado de las solicitudes asincrónicas.

> **Note**
>
> La consola solo muestra el estado de las solicitudes asincrónicas realizadas en los 14 días anteriores. Para ver el estado de solicitudes anteriores, utilice la AWS CLI, los SDK o la API REST.

Las operaciones de administración asíncrona pueden tener uno de los siguientes estados:

NEW

Amazon S3 ha recibido la solicitud y se está preparando para realizar la operación.

IN_PROGRESS

Amazon S3 está llevando a cabo la operación actualmente.

SUCCESS

La operación se ha completado correctamente. La respuesta incluye información relevante, como el alias de punto de acceso de varias regiones para una solicitud de `CreateMultiRegionAccessPoint`.

FAILED

La operación produce un error. La respuesta incluye un mensaje de error que indica el motivo del error de la solicitud.

Uso de AWS CloudTrail con puntos de acceso de varias regiones

Puede utilizar AWS CloudTrail para ver, buscar, descargar, archivar, analizar y responder a la actividad de la cuenta en la infraestructura de AWS. Con los puntos de acceso de varias regiones y el registro de CloudTrail, puede identificar lo siguiente:

- Quién o qué ha tomado qué medida
- Sobre qué recursos se ha actuado
- Cuándo se produjo el evento
- Otros detalles sobre el evento

Puede utilizar esta información de registro para analizar y responder a la actividad que se produjo a través de sus puntos de acceso de varias regiones.

Cómo configurar AWS CloudTrail para puntos de acceso de varias regiones

Para habilitar el registro de CloudTrail para cualquier operación relacionada con la creación o el mantenimiento de puntos de acceso de varias regiones, debe configurar el registro de CloudTrail para registrar los eventos en la región de Oeste de EE. UU. (Oregón). Debe configurar su registro de este modo, independientemente de la región en la que se encuentre al realizar la solicitud o de las regiones que admita el punto de acceso de varias regiones. Todas las solicitudes para crear o mantener un punto de acceso de varias regiones se dirigen a través de la región de Oeste de EE. UU. (Oregón). Le recomendamos que agregue esta región a un seguimiento existente o que cree un nuevo seguimiento que contenga esta región y todas las regiones asociadas al punto de acceso de varias regiones.

Amazon S3 registra todas las solicitudes realizadas a través de los puntos de acceso de varias regiones y las solicitudes realizadas a las operaciones de la API que administran puntos de acceso, tales como `CreateMultiRegionAccessPoint` y `GetMultiRegionAccessPointPolicy`. Cuando registra estas solicitudes a través de un punto de acceso de varias regiones, aparecen en los registros de AWS CloudTrail con el nombre de host del punto de acceso de varias regiones. Por ejemplo, si realiza solicitudes a un bucket a través de un punto de acceso de varias regiones con el alias `mfzwi23gnjvgw.mrap`, las entradas en el registro de CloudTrail tendrían un nombre de host de `mfzwi23gnjvgw.mrap.accesspoint.s3-global.amazonaws.com`.

Los puntos de acceso de varias regiones dirigen las solicitudes al bucket más cercano. Debido a este comportamiento, cuando busque los registros de CloudTrail para un punto de acceso de varias regiones, verá que las solicitudes se realizarán a los buckets subyacentes. Algunas de esas solicitudes pueden ser solicitudes directas al bucket que no se enrutan a través del punto de acceso de varias regiones. Es importante recordarlo al revisar el tráfico. Cuando un bucket se encuentra en un punto de acceso de varias regiones, las solicitudes se pueden realizar directamente a ese bucket sin pasar por el punto de acceso de varias regiones.

Hay eventos asíncronos relacionados con la creación y administración de puntos de acceso de varias regiones. Las solicitudes asincrónicas no tienen eventos de finalización en el registro de CloudTrail. Para obtener más información acerca de solicitudes asincrónicas, consulte Monitoreo y registro de solicitudes realizadas a las operaciones de la API de administración de puntos de acceso de varias regiones.

Para obtener más información sobre AWS CloudTrail, consulte ¿Qué es AWS CloudTrail? en la Guía del usuario de AWS CloudTrail.

Seguridad en Amazon S3

La seguridad en la nube de AWS es la mayor prioridad. Como cliente de AWS, se beneficiará de una arquitectura de red y un centro de datos que están diseñados para satisfacer los requisitos de seguridad de las organizaciones más exigentes.

La seguridad es una responsabilidad compartida entre AWS y usted. El modelo de responsabilidad compartida la describe como seguridad de la nube y seguridad en la nube:

Seguridad de la nube

AWS es responsable de proteger la infraestructura que ejecuta servicios de AWS en la Nube de AWS. AWS también le proporciona servicios que puede utilizar de forma segura. Auditores externos prueban y verifican periódicamente la eficacia de nuestra seguridad en el marco de los programas de conformidad de AWS. Para obtener más información acerca de los programas de conformidad que se aplican a Amazon S3, consulte Servicios de AWS en el ámbito del programa de conformidad.

Seguridad en la nube

Su responsabilidad la determina el servicio de AWS que utilice. Usted también es responsable de otros factores, incluida la confidencialidad de los datos, los requisitos de la empresa y la legislación y los reglamentos aplicables. Para Amazon S3, las siguientes áreas son su responsabilidad:

- Gestión de los datos, incluida la propiedad del objeto y el cifrado.
- Clasificación de sus activos
- Administración del acceso a los datos con roles de IAM y otras configuraciones de servicio para aplicar los permisos adecuados.
- Habilitación de controles detectives, como AWS CloudTrail o Amazon GuardDuty para Amazon S.

Esta documentación le ayudará a comprender cómo aplicar el modelo de responsabilidad compartida cuando se utiliza Amazon S3. En los siguientes temas, se le mostrará cómo configurar Amazon S3 para satisfacer sus objetivos de seguridad y conformidad. También aprenderá a utilizar otros servicios de AWS que lo pueden ayudar a monitorear y proteger sus recursos de Amazon S3.

> **Note**
>
> Para obtener más información sobre el uso de la clase de almacenamiento Amazon S3 Express One Zone con buckets de directorio, consulte ¿Qué es S3 Express One Zone? y Buckets de directorio.

Temas

Protección de los datos en Amazon S3

Amazon S3 proporciona una infraestructura de almacenamiento de alta durabilidad diseñada para el almacenamiento de datos principales y críticos. S3 Standard, S3 Intelligent-Tiering, S3 Standard-IA, S3 Glacier Instant Retrieval, S3 Glacier Flexible Retrieval y S3 Glacier Deep Archive almacenan objetos de forma redundante en varios dispositivos en un mínimo de tres zonas de disponibilidad en una Región de AWS. Una zona de disponibilidad consiste en uno o varios centros de datos discretos con alimentación, redes y conectividad redundantes en una Región de AWS. Las zonas de disponibilidad están separadas físicamente por una distancia significativa, muchos kilómetros, de cualquier otra zona de disponibilidad, aunque todas están a menos de 100 km (60 millas) entre sí. La clase de almacenamiento S3 One Zone-IA almacena datos de forma redundante en varios dispositivos dentro de una única zona de disponibilidad. Estos servicios están diseñados para gestionar los errores simultáneos de los dispositivos mediante la detección y la reparación rápidas de

cualquier pérdida de redundancia, y también verifican de forma periódica la integridad de los datos mediante sumas de comprobación.

El almacenamiento estándar de Amazon S3 ofrece las siguientes características:

- Respaldado por el Acuerdo de nivel de servicio de Amazon S3.
- Diseñado para ofrecer una durabilidad del 99,999999999 % y una disponibilidad de los objetos del 99,99 % durante un año concreto.
- S3 Standard, S3 Intelligent-Tiering, S3 Standard-IA, S3 Glacier Instant Retrieval, S3 Glacier Flexible Retrieval y S3 Glacier Deep Archive están diseñados para preservar datos en caso de pérdida de una zona de disponibilidad completa de Amazon S3.

Amazon S3 protege sus datos adicionalmente con el control de versiones. Puede utilizar el control de versiones para conservar, recuperar y restaurar todas las versiones de los objetos almacenados en su bucket de Amazon S3. Con el control de versiones, puede recuperarse fácilmente de acciones no deseadas del usuario y de errores de la aplicación. De forma predeterminada, las solicitudes recuperan la versión escrita más recientemente. Puede recuperar versiones más antiguas de un objeto especificando una versión del objeto en la solicitud.

Además del control de versiones de S3, también puede utilizar Amazon S3 Object Lock y la Replicación de S3 para proteger los datos. Para obtener más información, consulte Tutorial: Protecting data on Amazon S3 against accidental deletion or application bugs using S3 Versioning, S3 Object Lock, and S3 Replication (Protección de los datos en Amazon S3 contra la eliminación accidental o los errores en la aplicación mediante el control de versiones de S3, S3 Object Lock y la Replicación de S3).

Para fines de protección de datos, le recomendamos proteger las credenciales de la Cuenta de AWS y configurar cuentas de usuario individuales con AWS Identity and Access Management, de modo que a cada usuario se le concedan únicamente los permisos necesarios para llevar a cabo su trabajo.

Si necesita módulos criptográficos validados FIPS 140-2 al acceder a AWS a través de una interfaz de línea de comandos o una API, utilice un punto de conexión de FIPS. Para obtener más información acerca de los puntos de enlace de FIPS disponibles, consulte Estándar de procesamiento de la información federal (FIPS) 140-2.

Las siguientes prácticas recomendadas sobre seguridad también evalúan la protección de datos en 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

Protección de los datos mediante el cifrado

> **Important**
>
> Amazon S3 aplica ahora el cifrado del servidor con claves administradas por Amazon S3 (SSE-S3) como el nivel básico de cifrado para cada bucket de Amazon S3. Desde el 5 de enero de 2023, todas las cargas de objetos nuevos a Amazon S3 se cifran automáticamente sin costo adicional y sin afectar al rendimiento. El estado de cifrado automático para la configuración de cifrado predeterminada en el bucket de S3 y para cargas de objetos nuevos está disponible en registros de AWS CloudTrail, Inventario de S3, Lente de almacenamiento de S3, la consola de Amazon S3 y como encabezado de respuesta a la API de Amazon S3 adicional en AWS Command Line Interface y los SDK de AWS. Para obtener más información, consulte Preguntas frecuentes del cifrado predeterminado.

La protección de datos se refiere a salvaguardarlos mientras están en tránsito (al desplazarlos desde y hacia Amazon S3) y en reposo (almacenados en discos en centros de datos Amazon S3). Puede proteger los datos en tránsito con capa de sockets seguros/seguridad de la capa de transporte (SSL/TLS) o con cifrado del cliente. Para proteger los datos en reposo en Amazon S3, dispone de las siguientes opciones:

- Cifrado del lado del servidor: Amazon S3 cifra sus objetos antes de guardarlos en discos en centros de datos de AWS y, a continuación, los descifra cuando se descargan.

  Todos los buckets de Amazon S3 tienen el cifrado configurado de forma predeterminada y todos los objetos nuevos cargados en un bucket de S3 se cifran automáticamente en reposo. El cifrado del lado del servidor con claves administradas de Amazon S3 (SSE-S3) es la configuración de cifrado predeterminada para cada bucket de Amazon S3. Para usar otro tipo de cifrado, puede especificar el tipo de cifrado del servidor que se utilizará en las solicitudes PUT de S3 o puede establecer la configuración de cifrado predeterminada en el bucket de destino.

Si desea especificar un tipo de cifrado diferente en sus solicitudes PUT, puede utilizar el cifrado del servidor con claves de AWS Key Management Service (AWS KMS) (SSE-KMS), el cifrado del servidor de doble capa con claves de AWS KMS (DSSE-KMS) o el cifrado del servidor con claves proporcionadas por el cliente (SSE-C). Si desea establecer una configuración de cifrado predeterminada diferente en el bucket de destino, puede usar SSE-KMS o DSSE-KMS.

Para obtener más información acerca de cada opción para el cifrado del servidor, consulte Protección de los datos con el cifrado del servidor.

Para configurar el cifrado del servidor, consulte:

- Especificación del cifrado del servidor con claves administradas por Amazon S3 (SSE-S3)
- Especificación del cifrado del lado del servidor con AWS KMS (SSE-KMS)
- the section called “Especificación de DSSE-KMS”
- Especificación del cifrado del lado del servidor con claves proporcionadas por el cliente (SSE-C)

- Cifrado del cliente: puede cifrar datos del lado del cliente y cargar los datos cifrados en Amazon S3. En este caso, administra el proceso de cifrado, las claves de cifrado y las herramientas relacionadas.

  Para configurar el cifrado del lado del cliente, consulte Protección de los datos con el cifrado del cliente.

Para ver qué porcentaje de los bytes de almacenamiento están cifrados, puede utilizar las métricas de Lente de almacenamiento de Amazon S3. Lente de almacenamiento de S3 es una función de análisis de almacenamiento en la nube que puede utilizar para obtener visibilidad en toda la organización sobre el uso y la actividad del almacenamiento de objetos. Para obtener más información, consulte Evaluación de la actividad y el uso de almacenamiento con S3 Storage Lens. Para obtener una lista completa de las métricas, consulte el Glosario de métricas de Lente de almacenamiento de S3.

Para obtener más información sobre el cifrado del lado del servidor y el cifrado del cliente, revise los temas siguientes.

Temas

- Protección de los datos con el cifrado del servidor
- Protección de los datos con el cifrado del cliente

Protección de los datos con el cifrado del servidor

> **Important**
>
> Amazon S3 aplica ahora el cifrado del servidor con claves administradas por Amazon S3 (SSE-S3) como el nivel básico de cifrado para cada bucket de Amazon S3. Desde el 5 de enero de 2023, todas las cargas de objetos nuevos a Amazon S3 se cifran automáticamente sin costo adicional y sin afectar al rendimiento. El estado de cifrado automático para la configuración de cifrado predeterminada en el bucket de S3 y para cargas de objetos nuevos está disponible en registros de AWS CloudTrail, Inventario de S3, Lente de almacenamiento de S3, la consola de Amazon S3 y como encabezado de respuesta a la API de Amazon S3 adicional en AWS Command Line Interface y los SDK de AWS. Para obtener más información, consulte Preguntas frecuentes del cifrado predeterminado.

El cifrado del lado del servidor es el cifrado de datos en su destino por la aplicación o servicio que los recibe. Amazon S3 cifra sus datos en el nivel de objeto; los escribe en los discos de sus centros de datos de AWS y los descifra cuando accede a él. Siempre que autentique su solicitud y tenga permiso de acceso, no existe diferencia alguna en la forma de obtener acceso a objetos cifrados o sin cifrar. Por ejemplo, si comparte objetos con una URL prefirmada, esa URL funcionará igual para objetos cifrados y sin cifrar. Además, al enumerar los objetos en su bucket, las operaciones de la API de listado devuelven una lista de todos los objetos, independientemente de si están cifrados.

Todos los buckets de Amazon S3 tienen el cifrado configurado de forma predeterminada y todos los objetos nuevos cargados en un bucket de S3 se cifran automáticamente en reposo. El cifrado del lado del servidor con claves administradas de Amazon S3 (SSE-S3) es la configuración de cifrado predeterminada para cada bucket de Amazon S3. Para usar otro tipo de cifrado, puede especificar el tipo de cifrado del servidor que se utilizará en las solicitudes PUT de S3 o puede establecer la configuración de cifrado predeterminada en el bucket de destino.

Si desea especificar un tipo de cifrado diferente en sus solicitudes PUT, puede utilizar el cifrado del servidor con claves de AWS Key Management Service (AWS KMS) (SSE-KMS), el cifrado del servidor de doble capa con claves de AWS KMS (DSSE-KMS) o el cifrado del servidor con claves proporcionadas por el cliente (SSE-C). Si desea establecer una configuración de cifrado predeterminada diferente en el bucket de destino, puede usar SSE-KMS o DSSE-KMS.

> **Note**
>
> No es posible aplicar tipos diferentes de cifrado en el servidor al mismo objeto simultáneamente.

Si necesita cifrar los objetos existentes, utilice Operaciones por lotes de S3 e Inventario de S3. Para obtener más información, consulte Cifrado de objetos con Operaciones por lotes de Amazon S3 y Realización de operaciones por lotes a gran escala en objetos de Amazon S3.

Tiene cuatro opciones mutuamente excluyentes de cifrado del servidor en función de cómo elija administrar las claves de cifrado y el número de capas de cifrado que quiera aplicar.

Cifrado del servidor con claves administradas por Amazon S3 (SSE-S3)

Todos los buckets de Amazon S3 tienen el cifrado configurado de forma predeterminada. La opción predeterminada para el cifrado del lado del servidor son las claves administradas de Amazon S3 (SSE-S3) Cada objeto se cifra con una clave única. Como medida de seguridad adicional, SSE-S3 cifra la propia clave con una clave raíz que cambia periódicamente. SSE-S3 utiliza uno de los cifrados de bloques más seguros disponibles, Advanced Encryption Standard de 256 bits (AES-256), para cifrar sus datos. Para obtener más información, consulte Uso del cifrado del servidor con claves administradas por Amazon S3 (SSE-S3).

Cifrado del lado del servidor con claves AWS Key Management Service (AWS KMS) (SSE-KMS)

El cifrado del lado del servidor con AWS KMS keys (SSE-KMS) se proporciona mediante una integración del servicio de AWS KMS con Amazon S3. Con AWS KMS, tiene más control sobre sus claves. Por ejemplo, puede ver claves distintas, editar las políticas de control y seguir las claves en AWS CloudTrail. Además, puede crear y gestionar claves administradas por el cliente o utilizar Claves administradas por AWS que sean únicas para usted, su servicio y su región. Para obtener más información, consulte Uso del cifrado del servidor con claves de AWS KMS (SSE-KMS).

Cifrado del servidor con claves de AWS Key Management Service (AWS KMS) (SSE-KMS)

El cifrado del servidor de doble capa con AWS KMS keys (DSSE-KMS) es similar al de SSE-KMS, pero el DSSE-KMS aplica dos capas individuales de cifrado en el nivel de objeto en lugar de una. Como ambas capas de cifrado se aplican a un objeto del lado del servidor, puede utilizar una amplia gama de Servicios de AWS y herramientas para analizar los datos en S3 y, al mismo tiempo, utilizar un método de cifrado que pueda satisfacer sus requisitos de conformidad. Para obtener más

información, consulte Uso del cifrado del servidor de doble capa con claves de AWS KMS (DSSE-KMS).

Cifrado en el servidor con claves proporcionadas por el cliente (SSE-C)

Con el cifrado del servidor con claves proporcionadas por el cliente (SSE-C), usted administra las claves de cifrado y Amazon S3 administra tanto el cifrado, al escribir en los discos, como el descifrado, cuando usted accede a los objetos. Para obtener más información, consulte Uso de cifrado en el lado del servidor con claves proporcionadas por el cliente (SSE-C).

Amazon S3 cifra ahora de forma automática todos los objetos nuevos

Amazon S3 aplica ahora el cifrado del servidor con claves administradas por Amazon S3 (SSE-S3) como el nivel básico de cifrado para cada bucket de Amazon S3. Desde el 5 de enero de 2023, todas las cargas de objetos nuevos a Amazon S3 se cifran automáticamente sin costo adicional y sin afectar al rendimiento. SSE-S3, que utiliza el estándar de cifrado avanzado de 256 bits (AES-256), se aplica automáticamente a todos los buckets nuevos y a cualquier bucket de S3 existente que aún no tenga configurado el cifrado predeterminado. El estado de cifrado automático para la configuración de cifrado predeterminada en el bucket de S3 y para cargas de objetos nuevos está disponible en registros de AWS CloudTrail, Inventario de S3, Lente de almacenamiento de S3, la consola de Amazon S3 y como encabezado de respuesta a la API de Amazon S3 adicional en AWS Command Line Interface (AWS CLI) y los SDK de AWS.

Las siguientes secciones responden a las preguntas sobre esta actualización.

¿Cambiará Amazon S3 la configuración de cifrado predeterminada de los buckets que ya tienen el cifrado predeterminado configurado?

No. No se modificará la configuración de cifrado predeterminada para un bucket existente que ya tenga configurado el SSE-S3 o el cifrado del lado del servidor con claves de AWS Key Management Service (AWS KMS) (SSE-KMS). Para obtener más información sobre cómo definir el comportamiento de cifrado predeterminado para los bucket, consulte Establecer el comportamiento del cifrado predeterminado del lado del servidor para los buckets de Amazon S3. Para obtener más información sobre la configuración de cifrado SSE-S3 y SSE-KMS, consulte Protección de los datos con el cifrado del servidor.

¿Se habilitará el cifrado predeterminado en los buckets que no tengan configurado el cifrado predeterminado?

Sí. Amazon S3 ahora configura el cifrado predeterminado en todos los bucket no cifrados existentes para aplicar cifrado del lado del servidor con claves administradas de S3 (SSE-S3) como nivel base

de cifrado para los objetos nuevos cargados en estos buckets. Los objetos que ya estén en un bucket sin cifrar existente no se cifrarán automáticamente.

¿Cómo puedo ver el estado de cifrado predeterminado de las cargas de objetos nuevos?

Actualmente, puede ver el estado de cifrado predeterminado para cargas de objetos nuevos en registros de AWS CloudTrail, Inventario de S3, Lente de almacenamiento de S3, la consola de Amazon S3 y como encabezado de respuesta a la API de Amazon S3 adicional en AWS Command Line Interface (AWS CLI) y los SDK de AWS.

- Para ver los eventos de CloudTrail, consulte Visualizar eventos de CloudTrail Insights en la consola de CloudTrail en la Guía del usuario de AWS CloudTrail. Los registros de CloudTrail proporcionan un seguimiento de la API para solicitudes `PUT` y `POST` a Amazon S3. Cuando se utilice el cifrado predeterminado para cifrar los objetos de los buckets, los registros de CloudTrail para las solicitudes de la API `PUT` y `POST` incluirán el siguiente campo como par de nombre-valor: `"SSEApplied":"Default_SSE_S3"`.
- Para consultar el estado de cifrado automático de las cargas de objetos nuevos en el inventario de S3, configure un informe de inventario de S3 para incluir el campo de metadatos de Encryption (Cifrado) y, a continuación, consulte el estado de cifrado de cada objeto nuevo del informe. Para obtener más información, consulte Configuración de inventario de Amazon S3.
- Para consultar el estado del cifrado automático de las cargas de objetos nuevos en S3 Storage Lens, configure un panel de S3 Storage Lens y consulte las métricas de Encrypted bytes (Bytes cifrados) y Encrypted object count (Recuento de objetos cifrados) en la categoría Data protection (Protección de datos) del panel. Para obtener más información, consulte Creación de un panel de Amazon S3 Storage Lens y Visualización de las métricas de S3 Storage Lens en los paneles.
- Para ver el estado del cifrado automático en el nivel de bucket en la consola de Amazon S3, compruebe el cifrado predeterminado de los buckets de Amazon S3 en la consola de Amazon S3. Para obtener más información, consulte Configuración del cifrado predeterminado.
- Para ver el estado del cifrado automático como encabezado de respuesta de la API de Amazon S3 adicional en la AWS Command Line Interface (AWS CLI) y los SDK de AWS, compruebe el encabezado de respuesta `x-amz-server-side-encryption` cuando utilice las API de acción de objetos, como PutObject y GetObject.

¿Qué tengo que hacer para aprovechar este cambio?

No es necesario que realice ningún cambio en las aplicaciones existentes. Como el cifrado predeterminado está habilitado para todos los buckets, todos los objetos nuevos cargados en Amazon S3 se cifran automáticamente.

¿Puedo desactivar el cifrado de los nuevos objetos que se escriben en mi bucket?

No. SSE-S3 es el nuevo nivel base de cifrado que se aplica a todos los objetos nuevos que se cargan en el bucket. Ya no puede desactivar el cifrado para las cargas de objetos nuevas.

¿Se verán afectados mis cargos?

No. El cifrado predeterminado con SSE-S3 está disponible sin costo adicional. Se le facturará el almacenamiento, las solicitudes y otras características de Amazon S3 como se haría normalmente. Para información sobre precios, consulte Precios de Amazon S3.

¿Amazon S3 cifrará mis objetos existentes que no estén cifrados?

No. A partir del 5 de enero de 2023, Amazon S3 solo cifra automáticamente las cargas de objetos nuevos. Para cifrar objetos existentes, puede utilizar Operaciones por lotes de S3 para crear copias cifradas de los objetos. Estas copias cifradas retendrán los datos y el nombre del objeto existente y se cifrarán mediante las claves de cifrado que especifique. Para obtener más información, consulte Cifrado de objetos con operaciones por lotes de Amazon S3 en el blog de almacenamiento de AWS.

No habilité el cifrado de mis buckets antes de esta versión. ¿Debo cambiar la forma de acceder a los objetos?

No. El cifrado predeterminado con SSE-S3 cifra automáticamente los datos según se escriben en Amazon S3 y los descifra para usted cuando acceda a ellos. No hay ningún cambio en la forma de acceder a los objetos que se cifran automáticamente.

¿Debo cambiar la forma de acceder a los objetos de cifrado del cliente?

No. Todos los objetos cifrados del cliente que se cifran antes de cargarse en Amazon S3 llegan como objetos de texto cifrado a Amazon S3. Estos objetos tendrán ahora una capa adicional de cifrado de SSE-S3. Las cargas de trabajo que utilizan objetos cifrados del cliente no requerirán ningún cambio en los servicios de cliente ni en la configuración de autorización.

> **Note**
>
> Los usuarios de HashiCorp Terraform que no utilicen una versión actualizada del proveedor de AWS, es posible que vean una desviación inesperada después de crear nuevos buckets

de S3 sin una configuración de cifrado definida por el cliente. Para evitar esta desviación, actualice la versión del proveedor de AWS de Terraform a una de las siguientes versiones: cualquier versión de 4.x, 3.76.1 o 2.70.4.

Uso del cifrado del servidor con claves administradas por Amazon S3 (SSE-S3)

Important

Amazon S3 aplica ahora el cifrado del servidor con claves administradas por Amazon S3 (SSE-S3) como el nivel básico de cifrado para cada bucket de Amazon S3. Desde el 5 de enero de 2023, todas las cargas de objetos nuevos a Amazon S3 se cifran automáticamente sin costo adicional y sin afectar al rendimiento. El estado de cifrado automático para la configuración de cifrado predeterminada en el bucket de S3 y para cargas de objetos nuevos está disponible en registros de AWS CloudTrail, Inventario de S3, Lente de almacenamiento de S3, la consola de Amazon S3 y como encabezado de respuesta a la API de Amazon S3 adicional en AWS Command Line Interface y los SDK de AWS. Para obtener más información, consulte Preguntas frecuentes del cifrado predeterminado.

Todas las cargas de objetos nuevos en los buckets de Amazon S3 se cifran de forma predeterminada con cifrado del lado del servidor con claves administradas de Amazon S3 (SSE-S3).

El cifrado de lado de servidor protege los datos en reposo. Amazon S3 cifra cada objeto con una clave única. Como medida de seguridad adicional, cifra la propia clave con una clave que rota regularmente. El cifrado del servidor de Amazon S3 utiliza el modo Galois/Counter Mode (AES-GCM) estándar de cifrado avanzado de 256 bits para cifrar todos los objetos cargados.

No se aplican cargos adicionales por usar el cifrado del lado del servidor con claves administradas de Amazon S3 (SSE-S3). Sin embargo, las solicitudes para configurar la característica de cifrado predeterminadas generan cargos por solicitudes de Amazon S3 estándar. Para obtener información acerca de los precios, consulte Precios de Amazon S3.

Si necesita que las cargas de datos se cifren únicamente con claves administradas de Amazon S3, puede utilizar la siguiente política de buckets. Por ejemplo, en la siguiente política de bucket se deniega el permiso para cargar un objeto a menos que la solicitud incluya el encabezado `x-amz-server-side-encryption` para solicitar el cifrado del lado del servidor:

```
{
```

```
  "Version": "2012-10-17",
  "Id": "PutObjectPolicy",
  "Statement": [
    {
      "Sid": "DenyObjectsThatAreNotSSES3",
      "Effect": "Deny",
      "Principal": "*",
      "Action": "s3:PutObject",
      "Resource": "arn:aws:s3:::amzn-s3-demo-bucket/*",
      "Condition": {
        "StringNotEquals": {
          "s3:x-amz-server-side-encryption": "AES256"
        }
      }
    }
   ]
}
```

> **Note**
>
> El cifrado en el servidor solo cifra los datos de objetos, no los metadatos de objetos.

Compatibilidad con API para el cifrado del lado del servidor

Todos los buckets de Amazon S3 tienen el cifrado configurado de forma predeterminada y todos los objetos nuevos cargados en un bucket de S3 se cifran automáticamente en reposo. El cifrado del lado del servidor con claves administradas de Amazon S3 (SSE-S3) es la configuración de cifrado predeterminada para cada bucket de Amazon S3. Para usar otro tipo de cifrado, puede especificar el tipo de cifrado del servidor que se utilizará en las solicitudes PUT de S3 o puede establecer la configuración de cifrado predeterminada en el bucket de destino.

Si desea especificar un tipo de cifrado diferente en sus solicitudes PUT, puede utilizar el cifrado del servidor con claves de AWS Key Management Service (AWS KMS) (SSE-KMS), el cifrado del servidor de doble capa con claves de AWS KMS (DSSE-KMS) o el cifrado del servidor con claves proporcionadas por el cliente (SSE-C). Si desea establecer una configuración de cifrado predeterminada diferente en el bucket de destino, puede usar SSE-KMS o DSSE-KMS.

Para configurar el cifrado del lado del servidor con las API de REST de creación de objetos, debe proporcionar el encabezado de solicitud `x-amz-server-side-encryption`. Para obtener más información acerca de las API de REST, consulte Uso de la API de REST.

Las siguientes API de Amazon S3 admiten este encabezado:

- Operaciones PUT: especifique el encabezado de solicitud al cargar datos con la API PUT. Para obtener más información, consulte PUT Object.
- Iniciar carga multiparte: especifique el encabezado en la solicitud de inicio cuando cargue objetos grandes mediante la API de carga multiparte. Para obtener más información, consulte la sección sobre Cómo iniciar la carga multiparte.
- Operaciones COPY: cuando copia un objeto, tiene un objeto de origen y otro de destino. Para obtener más información, consulte Objeto PUT - Copia.

> **Note**
>
> Cuando utilice una operación POST para cargar un objeto, en vez de proporcionar el encabezado de solicitud, debe proporcionar la misma información en los campos del formulario. Para obtener más información, consulte POST Object.

Los AWS SDK también proporcionan API de encapsulamiento que puede utilizar para solicitar el cifrado del lado del servidor. También puede usar la AWS Management Console para cargar objetos y solicitar el cifrado del lado del servidor.

Para obtener más información, consulte Conceptos de AWS KMS en la Guía para desarrolladores de AWS Key Management Service.

Temas

- Especificación del cifrado del servidor con claves administradas por Amazon S3 (SSE-S3)

Especificación del cifrado del servidor con claves administradas por Amazon S3 (SSE-S3)

> **Important**
>
> Amazon S3 aplica ahora el cifrado del servidor con claves administradas por Amazon S3 (SSE-S3) como el nivel básico de cifrado para cada bucket de Amazon S3. Desde el 5 de enero de 2023, todas las cargas de objetos nuevos a Amazon S3 se cifran automáticamente sin costo adicional y sin afectar al rendimiento. El estado de cifrado automático para la configuración de cifrado predeterminada en el bucket de S3 y para cargas de objetos nuevos está disponible en registros de AWS CloudTrail, Inventario de S3, Lente de almacenamiento

de S3, la consola de Amazon S3 y como encabezado de respuesta a la API de Amazon S3 adicional en AWS Command Line Interface y los SDK de AWS. Para obtener más información, consulte Preguntas frecuentes del cifrado predeterminado.

Todos los buckets de Amazon S3 tienen el cifrado configurado de forma predeterminada y todos los objetos nuevos cargados en un bucket de S3 se cifran automáticamente en reposo. El cifrado del lado del servidor con claves administradas de Amazon S3 (SSE-S3) es la configuración de cifrado predeterminada para cada bucket de Amazon S3. Para usar otro tipo de cifrado, puede especificar el tipo de cifrado del servidor que se utilizará en las solicitudes PUT de S3 o puede establecer la configuración de cifrado predeterminada en el bucket de destino.

Si desea especificar un tipo de cifrado diferente en sus solicitudes PUT, puede utilizar el cifrado del servidor con claves de AWS Key Management Service (AWS KMS) (SSE-KMS), el cifrado del servidor de doble capa con claves de AWS KMS (DSSE-KMS) o el cifrado del servidor con claves proporcionadas por el cliente (SSE-C). Si desea establecer una configuración de cifrado predeterminada diferente en el bucket de destino, puede usar SSE-KMS o DSSE-KMS.

Puede especificar SSE-S3 mediante la consola de S3, las API de REST, los SDK de AWS y la AWS Command Line Interface (AWS CLI). Para obtener más información, consulte Establecer el comportamiento del cifrado predeterminado del lado del servidor para los buckets de Amazon S3.

Uso de la consola de S3

En este tema se describe cómo configurar o cambiar el tipo de cifrado que utiliza un objeto mediante la AWS Management Console. Cuando copia un objeto con la consola, Amazon S3 copia el objeto tal cual. Esto significa que si el objeto de origen está cifrado, el objeto de destino también lo está. También puede usar la consola para agregar o cambiar el cifrado de un objeto.

Note

- Si cambia el cifrado de un objeto, se crea un nuevo objeto para reemplazar el antiguo. Si el control de versiones de S3 está activado, se crea una nueva versión del objeto y el objeto existente se convierte en una versión anterior. El rol que cambia la propiedad también se convierte en el propietario del nuevo objeto o (versión del objeto).
- Si cambia el tipo de cifrado de un objeto que tiene etiquetas definidas por el usuario, debe tener el permiso `s3:GetObjectTagging`. Si va a cambiar el tipo de cifrado de un objeto

que no tiene etiquetas definidas por el usuario pero que tiene un tamaño superior a 16 MB, también debe tener el permiso `s3:GetObjectTagging`.

Si la política de bucket de destino deniega la acción `s3:GetObjectTagging`, el tipo de cifrado del objeto se actualizará, pero las etiquetas definidas por el usuario se eliminarán del objeto y aparecerá un error.

Para cambiar el cifrado de un objeto

1. Inicie sesión AWS Management Console Management Console y abra la consola de Amazon S3 en https://console.aws.amazon.com/s3/.
2. En el panel de navegación izquierdo, elija Instancias.
3. En la lista Buckets (Buckets), elija el nombre del bucket que contiene el objeto.
4. En la lista Objects (Objetos), seleccione el nombre del objeto al que desea agregar cifrado o cuyo cifrado desea modificar.

   Aparece la página de detalles del objeto, con varias secciones que muestran las propiedades del objeto.
5. Elija la pestaña Propiedades.
6. Desplácese hacia abajo hasta la sección Configuración del cifrado del lado del servidor y, a continuación, seleccione Editar.
7. En Configuración del cifrado, elija Usar la configuración del bucket para el cifrado predeterminado o Anular la configuración del bucket para el cifrado predeterminado.
8. Si elige Anular la configuración del bucket para el cifrado predeterminado, debe configurar los siguientes ajustes de cifrado.

   - Para el Tipo de cifrado, elija las Claves administradas de Amazon S3 (SSE-S3). SSE-S3 utiliza uno de los cifrados de bloques más seguros, Advanced Encryption Standard de 256 bits (AES-256), para cifrar cada objeto. Para obtener más información, consulte Uso del cifrado del servidor con claves administradas por Amazon S3 (SSE-S3).
9. Elija Save changes (Guardar cambios).

> **Note**
>
> En esta acción se aplica el cifrado a todos los objetos especificados. Al cifrar carpetas, espere a que finalice la operación de guardado para agregar nuevos objetos a la carpeta.

Uso de la API de REST

Al crear un objeto, es decir, cuando carga un objeto nuevo o hace una copia de un objeto existente, puede especificar si desea que Amazon S3 cifre los datos con claves administradas de Amazon S3 (SSE-S3) al agregar el encabezado `x-amz-server-side-encryption` en la solicitud. Configure el valor del encabezado para el algoritmo de cifrado `AES256` que admite Amazon S3. Amazon S3 confirma que su objeto se ha almacenado con SSE-S3 al devolver el encabezado de respuesta `x-amz-server-side-encryption`.

Las siguientes operaciones de la API de carga de REST aceptan el encabezado de solicitud `x-amz-server-side-encryption`.

- PUT Object
- PUT Object - Copy
- POST Object
- Initiate Multipart Upload

Cuando cargue objetos grandes con la operación de la API de carga multiparte, puede especificar el cifrado del lado del servidor añadiendo el encabezado `x-amz-server-side-encryption` a la solicitud Iniciar carga multiparte. Cuando copie un objeto existente, independientemente de si el objeto de origen está cifrado o no, el objeto de destino no estará cifrado, a no ser que solicite explícitamente el cifrado del lado del servidor.

Los encabezados de respuesta de las siguientes operaciones de la API de REST devuelven el encabezado `x-amz-server-side-encryption` cuando un objeto se almacena con SSE-S3.

- PUT Object
- PUT Object - Copy
- POST Object
- Initiate Multipart Upload
- Upload Part

- Upload Part - Copy
- Complete Multipart Upload
- Get Object
- Head Object

> **Note**
>
> No envíe encabezados de solicitud de cifrado para las solicitudes `GET` y `HEAD` si el objeto utiliza SSE-S3, porque recibirá un error de código de error HTTP 400 (Solicitud errónea).

Uso de los AWS SDK

Si utiliza los SDK de AWS, puede solicitar a Amazon S3 que utilice el cifrado del lado del servidor con claves de cifrado administradas por Amazon S3 (SSE-S3). En esta sección, se proporcionan ejemplos de uso de los AWS SDK en diferentes lenguajes. Para obtener información acerca de otros SDK, consulte Código de muestra y bibliotecas.

Java

Cuando use el AWS SDK for Java para cargar un objeto, puede usar SSE-S3 para cifrarlo. Para solicitar cifrado del lado del servidor utilice la propiedad `ObjectMetadata` de la `PutObjectRequest` para establecer el encabezado de solicitud `x-amz-server-side-encryption`. Cuando llama al método `putObject()` del `AmazonS3Client`, Amazon S3 cifra y guarda los datos.

También puede solicitar el cifrado SSE-S3 cuando cargue objetos con la operación de la API de carga multiparte:

- Al usar la operación de la API de carga multiparte de nivel alto, utiliza los métodos `TransferManager` para aplicar cifrado del lado del servidor a los objetos a medida que los carga. Puede utilizar cualquier método de carga que tome `ObjectMetadata` como parámetro. Para obtener más información, consulte Carga de un objeto con la carga multiparte.
- Cuando utiliza la operación de la API de carga multiparte de nivel bajo, especifica el cifrado del lado del servidor al iniciar la carga multiparte. Añade la propiedad `ObjectMetadata` al llamar al método `InitiateMultipartUploadRequest.setObjectMetadata()`. Para obtener más información, consulte Uso de los AWS SDK (API de bajo nivel).

No puede cambiar directamente el estado de cifrado de un objeto (cifrado de un objeto no cifrado o descifrado de un objeto cifrado). Para cambiar el estado de cifrado de un objeto, realice una copia del objeto, especifique el estado de cifrado deseado para la copia y elimine el objeto original. Amazon S3 cifra el objeto copiado solo si solicita explícitamente el cifrado del lado del servidor. Para solicitar el cifrado del objeto copiado por medio de la API de Java, use la propiedad `ObjectMetadata` para especificar el cifrado del lado del servidor en la `CopyObjectRequest`.

Example Ejemplo

En el siguiente ejemplo se muestra cómo establecer el cifrado del lado del servidor con el AWS SDK for Java. Se muestra cómo realizar las siguientes tareas:

- Cargue un objeto nuevo mediante SSE-S3.
- Cambiar el estado de cifrado de un objeto (en este ejemplo, cifrar un objeto que no estaba cifrado anteriormente) copiando el objeto.
- Comprobar el estado de cifrado del objeto.

Para obtener más información acerca del cifrado del lado del servidor, consulte Uso de la API de REST. Para obtener instrucciones sobre cómo crear y probar una muestra funcional, consulte Introducción en la Guía del desarrollador de AWS SDK for Java.

```
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

Cuando carga un objeto, puede indicar a Amazon S3 que lo cifre. Para cambiar el estado de cifrado de un objeto existente, realice una copia del objeto y elimine el objeto de origen. De forma predeterminada la operación de copia cifra el destino solo si usted solicita explícitamente cifrado del lado del servidor del objeto de destino. Para especificar SSE-S3 en el `CopyObjectRequest`, añada lo siguiente:

```
ServerSideEncryptionMethod = ServerSideEncryptionMethod.AES256
```

Para obtener una muestra funcional que indica cómo copiar un objeto, consulte Uso de los AWS SDK.

En el siguiente ejemplo se carga un objeto. En la solicitud, el ejemplo indica a Amazon S3 que cifre el objeto. A continuación el ejemplo recupera los metadatos de los objetos y comprueba el método de cifrado que se utilizó. Para obtener información acerca de cómo configurar y ejecutar ejemplos de código, consulte Introducción al SDK de AWS para .NET en la Guía para desarrolladores del SDK de AWS para .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

En este tema se explica cómo usar las clases de la versión 3 de AWS SDK for PHP para agregar SSE-S3 a objetos que cargue a Amazon S3. Para obtener más información acerca de la API del SDK de AWS para Ruby, consulte SDK de AWS para Ruby, versión 2.

Para cargar un objeto en Amazon S3, use el método Aws\S3\S3Client::putObject(). Para añadir el encabezado de solicitud `x-amz-server-side-encryption` a su solicitud de carga, especifique el parámetro `ServerSideEncryption` con el valor `AES256`, como se muestra en el siguiente ejemplo de código. Para obtener información acerca de solicitudes de cifrado del lado del servidor, consulte Uso de la API de 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',
]);
```

Como respuesta, Amazon S3 devuelve el encabezado `x-amz-server-side-encryption` con el valor del algoritmo de cifrado que se utilizó para cifrar los datos del objeto.

Cuando carga objetos grandes con la operación de la API de carga multiparte, puede especificar SSE-S3 para los objetos que está cargando, como sigue:

- Cuando utilice la operación de la API de carga multiparte de bajo nivel, especifique el cifrado del lado del servidor al llamar al método Aws\S3\S3Client::createMultipartUpload(). Para agregar el encabezado de solicitud `x-amz-server-side-encryption` a su solicitud, especifique la clave del parámetro de la `array` `ServerSideEncryption` con el valor `AES256`. Para obtener más información sobre la operación de la API de carga multiparte de bajo nivel, consulte Uso de los AWS SDK (API de bajo nivel).
- Cuando utilice la operación de la API de carga multiparte de alto nivel, especifique el cifrado del lado del servidor mediante el parámetro `ServerSideEncryption` de la operación de la API CreateMultipartUpload. Para ver un ejemplo sobre cómo usar el método `setOption()` con la operación de la API de carga multiparte de alto nivel, consulte Carga de un objeto con la carga multiparte.

Para determinar el estado de cifrado de un objeto existente, recupere los metadatos del objeto llamando al método Aws\S3\S3Client::headObject() como se muestra en el siguiente ejemplo de código PHP.

```
 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'];
```

Para cambiar el estado de cifrado de un objeto existente, realice una copia del objeto con el método Aws\S3\S3Client::copyObject() y elimine el objeto de origen. De forma predeterminada, `copyObject()` no cifra el objeto de destino, a menos que solicite de forma explícita el cifrado del lado del servidor con el parámetro `ServerSideEncryption` con el valor `AES256`. El siguiente ejemplo de código PHP realiza una copia de un objeto y agrega cifrado en el servidor al objeto copiado.

```
 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',
]);
```

Para obtener más información, consulte los temas siguientes:

- AWS SDK for PHP para la clase Aws\S3\S3Client de Amazon S3
- Documentación de AWS SDK for PHP

Ruby

Si utiliza AWS SDK for Ruby para cargar un objeto, puede especificar que el objeto se almacene cifrado en reposo con SSE-S3. Cuando vuelve a leer el objeto, este se descifra automáticamente.

En el siguiente ejemplo de AWS SDK for Ruby versión 3 se muestra cómo especificar que un archivo cargado en Amazon S3 quede cifrado en reposo.

```
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__
```

El siguiente ejemplo de código muestra cómo determinar el estado de cifrado de un objeto existente.

```
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 no se utiliza el cifrado del lado del servidor para el objeto almacenado en Amazon S3, el método devolverá `null`.

Para cambiar el estado de cifrado de un objeto existente, realice una copia del objeto y elimine el objeto de origen. De forma predeterminada, los métodos de copia no cifran el objeto de destino, a menos que solicite de forma explícita el cifrado del lado del servidor. Puede solicitar el cifrado del objeto de destino al especificar el valor `server_side_encryption` en el argumento hash de la opción, tal como se muestra en el siguiente código de ejemplo de Ruby. El ejemplo de código muestra cómo copiar un objeto y cifrar la copia con 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__
```

Uso de la AWS CLI

Para especificar SSE-S3 al cargar un objeto mediante la AWS CLI, utilice el siguiente ejemplo.

```
aws s3api put-object --bucket amzn-s3-demo-bucket1 --key object-key-name --server-side-
encryption AES256  --body file path
```

Para obtener más información, consulte put-object en la Referencia de la AWS CLI. Para especificar SSE-S3 al copiar un objeto mediante la AWS CLI, consulte copy-object.

Uso de AWS CloudFormation

Para obtener ejemplos de configuración de cifrado mediante AWS CloudFormation, consulte Crear un bucket con cifrado predeterminado y Crear un bucket mediante el cifrado del lado del servidor AWS KMS con una clave de bucket de S3 en el tema `AWS::S3::Bucket ServerSideEncryptionRule` de la Guía del usuario de AWS CloudFormation.

Uso del cifrado del servidor con claves de AWS KMS (SSE-KMS)

> **Important**
>
> Amazon S3 aplica ahora el cifrado del servidor con claves administradas por Amazon S3 (SSE-S3) como el nivel básico de cifrado para cada bucket de Amazon S3. Desde el 5 de enero de 2023, todas las cargas de objetos nuevos a Amazon S3 se cifran automáticamente sin costo adicional y sin afectar al rendimiento. El estado de cifrado automático para la configuración de cifrado predeterminada en el bucket de S3 y para cargas de objetos nuevos está disponible en registros de AWS CloudTrail, Inventario de S3, Lente de almacenamiento de S3, la consola de Amazon S3 y como encabezado de respuesta a la API de Amazon S3 adicional en AWS Command Line Interface y los SDK de AWS. Para obtener más información, consulte Preguntas frecuentes del cifrado predeterminado.

El cifrado del lado del servidor es el cifrado de datos en su destino por la aplicación o servicio que los recibe.

Amazon S3 habilita automáticamente el cifrado del lado del servidor con claves administradas de Amazon S3 (SSE-S3) para cargar objetos nuevos.

A menos que especifique lo contrario, los buckets utilizan SSE-S3 de forma predeterminada para cifrar objetos. Sin embargo, puede elegir configurar buckets para utilizar el cifrado del lado del servidor con claves de AWS Key Management Service (AWS KMS) (SSE-KMS). Para obtener más información, consulte Especificación del cifrado del lado del servidor con AWS KMS (SSE-KMS).

AWS KMS es un servicio que combina hardware y software seguros y de alta disponibilidad para ofrecer un sistema de administración de claves adaptado a la nube. Amazon S3 utiliza el cifrado del lado del servidor con AWS KMS (SSE-KMS) para cifrar los datos de objetos de S3. Además, cuando se solicita SSE-KMS para el objeto, la suma de comprobación de S3 (como parte de los metadatos del objeto) se almacena de forma cifrada. Para obtener más información acerca de la suma de comprobación, consulte Comprobación de la integridad de objetos.

Si utiliza claves KMS, puede utilizar AWS KMS en toda la AWS Management Console o la API de AWS KMS para hacer lo siguiente:

- Cree, vea, edite, monitoree, habilite o desactive, rote y programe la eliminación de claves de KMS de forma centralizada.

- Defina las políticas que controlan cómo y quién puede utilizar las claves KMS.
- Auditar su uso para demostrar que se están utilizando correctamente. Las auditorías están admitidas por la API de AWS KMS, pero no por la AWS Management Console de AWS KMS.

Los controles de seguridad de AWS KMS pueden ayudarlo a cumplir los requisitos de conformidad relacionados con el cifrado. Puede usar estas claves de KMS para proteger sus datos en buckets de Amazon S3. Al utilizar el cifrado SSE-KMS con un bucket de S3, AWS KMS keys debe estar en la misma región que ese bucket.

La utilización de AWS KMS keys conlleva cargos adicionales. Para obtener más información, consulte AWS KMS key los conceptos en la AWS Key Management Service Guía para desarrolladores y AWS KMS los precios.

Permisos

Para cargar un objeto cifrado con una AWS KMS key en Amazon S3, es necesario contar con los permisos `kms:GenerateDataKey` en la clave. Para descargar un objeto cifrado con una AWS KMS key, es necesario contar con los permisos `kms:Decrypt`. Para obtener más información sobre los permisos AWS KMS necesarios para las cargas multiparte, consulte API y permisos de carga multiparte.

> **Important**
>
> Revise detenidamente los permisos que se otorgan en sus políticas de claves de KMS. Limite siempre los permisos de la política de claves de KMS administradas por el cliente únicamente a las entidades principales de IAM y los servicios de AWS que deben acceder a la acción clave de AWS KMS correspondiente. Para más información, consulte Políticas de claves en AWS KMS.

Temas

- AWS KMS keys
- Claves de bucket de Amazon S3
- Requisito de cifrado del lado del servidor
- Contexto de cifrado

- Envío de solicitudes para objetos cifrados de AWS KMS
- Especificación del cifrado del lado del servidor con AWS KMS (SSE-KMS)
- Reducción del costo de SSE-KMS con las claves de bucket de Amazon S3

AWS KMS keys

Cuando utiliza el cifrado del lado del servidor con AWS KMS (SSE-KMS), puede utilizar la clave administrada por AWS predeterminada o especificar una clave administrada por el cliente que ya haya creado. AWS KMS admite el cifrado de sobres. S3 utiliza las funciones de AWS KMS para cifrado de sobres para proteger aún más los datos. El cifrado de sobres es la práctica de cifrar los datos que son texto no cifrado con una clave de datos y, a continuación, cifrar la propia clave de datos con una clave KMS. Para obtener más información acerca del cifrado de sobre, consulte Cifrado de sobre en la guía para desarrolladores de AWS Key Management Service.

Si no especifica una clave administrada por el cliente, Amazon S3 creará de manera automática una Clave administrada de AWS en su Cuenta de AWS la primera vez que agregue un objeto cifrado con SSE-KMS a un bucket. De forma predeterminada, Amazon S3 utiliza esta clave de KMS para SSE-KMS.

> **Note**
>
> Los objetos cifrados mediante SSE-KMS con Claves administradas por AWS no se pueden compartir entre cuentas. Si necesita compartir datos de SSE-KMS entre cuentas, debe utilizar una clave administrada por el cliente de AWS KMS.

Si desea utilizar una clave administrada por el cliente para SSE-KMS, cree una clave administrada por el cliente de cifrado simétrico antes de configurar SSE-KMS. Luego cuando configure SSE-KMS para el bucket, especifique la clave administrada por el cliente existente. Para obtener más información sobre la clave de cifrado simétrica, consulte Symmetric encryption KMS keys (Claves de KMS de cifrado simétricas) en la Guía para desarrolladores de AWS Key Management Service.

Crear una clave administrada por el cliente le da más flexibilidad y control. Por ejemplo, puede crear, rotar y deshabilitar las claves administradas por el cliente. También puede definir controles de acceso y auditar las claves administradas por el cliente que utiliza para proteger sus datos. Para obtener más información acerca de las claves administradas por el cliente y AWS, consulte Claves de cliente y claves de AWS en la Guía para desarrolladores de AWS Key Management Service.

> **Note**
>
> Cuando utiliza el cifrado del lado del servidor con una clave administrada por el cliente que se almacena en un almacén de claves externo, a diferencia de las claves de KMS estándar, es responsable de garantizar la disponibilidad y la durabilidad del material de la clave. Para obtener más información sobre los almacenes de claves externos y cómo cambian el modelo de responsabilidad compartida, consulte Almacenes de claves externos en la Guía para desarrolladores de AWS Key Management Service.

Uso del cifrado SSE-KMS para operaciones entre cuentas

Tenga en cuenta lo siguiente cuando utilice el cifrado para operaciones entre cuentas:

- Si no se proporciona un nombre de recurso de Amazon (ARN) o un alias de AWS KMS key en el momento de la solicitud, ni a través de la configuración de cifrado predeterminado del bucket, se usa la Clave administrada de AWS (`aws/s3`).
- Si está cargando o accediendo a objetos de S3 usando las entidades principales de AWS Identity and Access Management (IAM) que están en la misma Cuenta de AWS que la clave de KMS, puede usar la Clave administrada de AWS (`aws/s3`).
- Use una clave administrada por el cliente si desea conceder acceso entre cuentas a sus objetos de S3. Puede configurar la política de una clave administrada por el cliente para permitir el acceso desde otra cuenta.
- Si especifica una clave de KMS administrada por el cliente, le recomendamos que use un ARN totalmente cualificado de la clave de KMS. Si, en su lugar, utiliza un alias de clave de KMS, AWS KMS resolverá la clave dentro de la cuenta del solicitante. Esto puede dar como resultado datos cifrados con una clave de KMS que pertenece al solicitante y no al propietario del bucket.
- Debe especificar una clave para la que el solicitante le haya concedido permiso `Encrypt`. Para obtener más información, consulte Permitir a los usuarios de claves utilizar una clave de KMS para las operaciones criptográficas en la Guía para desarrolladores de AWS Key Management Service.

Para obtener más información acerca de cuándo utilizar claves administradas por el cliente y las claves de KMS administradas por AWS, consulte ¿Debo usar una clave administrada por Clave administrada de AWS o una clave administrada por el cliente para cifrar mis objetos en Amazon S3?

Flujo de trabajo de cifrado SSE-KMS

Si elige cifrar los datos mediante una Clave administrada de AWS o una clave administrada por el cliente, AWS KMS y Amazon S3 llevan a cabo las siguientes acciones de cifrado de sobre:

1. Amazon S3 solicita una clave de datos en texto no cifrado y una copia de la clave cifrada con la clave de KMS especificada.
2. AWS KMS crea una clave de datos, la cifra con la clave KMS y envía la clave de datos en texto no cifrado y la clave de datos cifrada a Amazon S3.
3. Amazon S3 cifra los datos con la clave de datos y elimina la clave en texto no cifrado de la memoria tan pronto como sea posible después de utilizarla.
4. Amazon S3 almacena la clave de datos cifrada como metadatos con el archivo de datos.

Cuando se solicita que se descifren los datos, Amazon S3 y AWS KMS realizan las siguientes acciones:

1. Amazon S3 envía la clave de datos cifrada a AWS KMS en una solicitud de `Decrypt`.
2. AWS KMS descifra la clave de datos mediante la misma clave KMS y devuelve la clave de datos en texto no cifrado a Amazon S3.
3. Amazon S3 descifra los datos cifrados, mediante la clave de datos de texto no cifrado y elimina la clave de datos de texto no cifrado de la memoria tan pronto como sea posible.

> **Important**
>
> Cuando utilice una AWS KMS key para el cifrado en el lado del servidor en Amazon S3, debe elegir una clave de cifrado de KMS simétrica. Amazon S3 solo admite claves KMS de cifrado simétricas. Para obtener más información sobre estas claves, consulte Symmetric encryption KMS keys (Claves de KMS de cifrado simétricas) en la Guía para desarrolladores de AWS Key Management Service.

Auditoría del cifrado SSE-KMS

Para identificar las solicitudes que especifican SSE-KMS, puede utilizar las métricas de Todas las solicitudes de SSE-KMS y Porcentaje de todas las solicitudes de SSE-KMS de SSE-KMS en las métricas de Lente de almacenamiento de Amazon S3. Lente de almacenamiento de S3 es una función de análisis de almacenamiento en la nube que puede utilizar para obtener visibilidad en toda

la organización sobre el uso y la actividad del almacenamiento de objetos. También puede utilizar el recuento de buckets con SSE-KMS habilitado y el porcentaje de buckets con SSE-KMS habilitado para comprender el recuento de los buckets (SSE-KMS) para el cifrado de buckets predeterminado. Para obtener más información, consulte Evaluación de la actividad y el uso de almacenamiento con S3 Storage Lens. Para obtener una lista completa de las métricas, consulte el Glosario de métricas de Lente de almacenamiento de S3.

Para auditar el uso de las claves de AWS KMS para los datos cifrados con SSE-KMS, puede utilizar los registros de AWS CloudTrail. Puede obtener información sobre las operaciones de cifrado, como, por ejemplo, GenerateDataKey y Decrypt. CloudTrail admite numerosos valores de atributos para filtrar la búsqueda, incluidos el nombre del evento, el nombre de usuario y la fuente del evento.

Claves de bucket de Amazon S3

Cuando configure el cifrado del lado del servidor mediante AWS KMS (SSE-KMS), puede configurar los buckets para que utilicen las claves de bucket de S3 para SSE-KMS. Usar una clave de nivel de bucket para SSE-KMS puede reducir los costos de solicitud de AWS KMS hasta en un 99 %, ya que disminuye el tráfico de solicitudes de Amazon S3 a AWS KMS.

Cuando configura un bucket para utilizar claves de bucket de S3 para SSE-KMS en objetos nuevos, AWS KMS genera una clave de bucket que se utiliza en la creación de claves de datos únicas para los objetos en el bucket. Esta clave de bucket de S3 se utiliza durante un periodo limitado dentro de Amazon S3, lo que reduce aún más la necesidad de que Amazon S3 realice solicitudes a AWS KMS para completar las operaciones de cifrado. Para obtener más información sobre el uso de claves de bucket de S3, consulte Reducción del costo de SSE-KMS con las claves de bucket de Amazon S3.

Requisito de cifrado del lado del servidor

Para requerir el cifrado del lado del servidor para todos los objetos en un bucket de Amazon S3 particular, puede usar una política de bucket. Por ejemplo, la siguiente política de bucket deniega el permiso de carga de objeto (`s3:PutObject`) para todos, si la solicitud no incluye el encabezado `x-amz-server-side-encryption-aws-kms-key-id`, que solicita el cifrado del lado del servidor con SSE-KMS.

```
{
   "Version":"2012-10-17",
   "Id":"PutObjectPolicy",
   "Statement":[{
         "Sid":"DenyObjectsThatAreNotSSEKMS",
         "Effect":"Deny",
         "Principal":"*",
```

```
         "Action":"s3:PutObject",
         "Resource":"arn:aws:s3:::amzn-s3-demo-bucket1/*",
         "Condition":{
            "Null":{
               "s3:x-amz-server-side-encryption-aws-kms-key-id":"true"
            }
         }
      }
   ]
}
```

Para requerir que una AWS KMS key particular se utilice para cifrar los objetos en un bucket, puede utilizar la clave de condición `s3:x-amz-server-side-encryption-aws-kms-key-id`. Para especificar la clave de KMS, debe utilizar un nombre de recurso de Amazon (ARN) que tenga el formato `arn:aws:kms:region:acct-id:key/key-id`. AWS Identity and Access Management no valida si existe la cadena para `s3:x-amz-server-side-encryption-aws-kms-key-id`.

> **Note**
>
> Al cargar un objeto, puede especificar la clave de KMS con el encabezado `x-amz-server-side-encryption-aws-kms-key-id` o confiar en la configuración de cifrado de bucket predeterminada. Si su solicitud PutObject especifica `aws:kms` en el encabezado `x-amz-server-side-encryption`, pero no especifica el encabezado `x-amz-server-side-encryption-aws-kms-key-id`, Amazon S3 asumirá que quiere usar la Clave administrada de AWS. En cualquier caso, el ID de la clave de AWS KMS que utiliza Amazon S3 para el cifrado de objetos debe coincidir con el ID de la clave de AWS KMS en la política; de lo contrario, Amazon S3 denegará la solicitud.

Para obtener una lista completa de las claves de condición específicas de Amazon S3, consulte Condition keys for Amazon S3 en la Referencia de autorizaciones de servicio.

Contexto de cifrado

Un contexto de cifrado es un conjunto definido de pares clave-valor que contienen información contextual adicional sobre los datos. El contexto de cifrado no está cifrado. Cuando se especifica un contexto de cifrado para una operación de cifrado, Amazon S3 debe especificar el mismo contexto de cifrado que para la operación de descifrado. De lo contrario, se produce un error en el descifrado.

AWS KMS utiliza el contexto de cifrado como datos autenticados adicionales (AAD) para admitir el cifrado autenticado. Para obtener más información sobre el contexto de cifrado, consulte Contexto de cifrado en la Guía para desarrolladores de AWS Key Management Service.

De forma predeterminada, Amazon S3 utiliza el nombre de recurso de Amazon (ARN) del objeto o bucket como par del contexto de cifrado:

- Si utiliza SSE-KMS sin activar una clave de bucket de S3, se utiliza el ARN del objeto como contexto de cifrado.

  ```
  arn:aws:s3:::object_ARN
  ```

- Si utiliza SSE-KMS y activa una clave de bucket de S3, se utiliza el ARN del bucket como contexto de cifrado. Para obtener más información sobre las claves de bucket de S3, consulte Reducción del costo de SSE-KMS con las claves de bucket de Amazon S3.

  ```
  arn:aws:s3:::bucket_ARN
  ```

Si lo desea, puede proporcionar un par de contexto de cifrado adicional mediante el encabezado `x-amz-server-side-encryption-context` en una solicitud s3:PutObject. No obstante, dado que el contexto de cifrado no está cifrado, asegúrese de no incluir información confidencial. Amazon S3 almacena este par de claves adicional junto con el contexto de cifrado predeterminado. Cuando procesa la solicitud PUT, Amazon S3 agrega el contexto de cifrado predeterminado de `aws:s3:arn` al que se proporcione.

Puede utilizar el contexto de cifrado para identificar y clasificar las operaciones criptográficas. También puede utilizar el valor ARN del contexto de cifrado predeterminado para realizar un seguimiento de las solicitudes relevantes en AWS CloudTrail consultando qué ARN de Amazon S3 se usó con qué clave de cifrado.

En el campo `requestParameters` de un archivo de registro de CloudTrail, el contexto de cifrado es similar al siguiente:

```
"encryptionContext": {
    "aws:s3:arn": "arn:aws:s3:::amzn-s3-demo-bucket1/file_name"
}
```

Cuando utiliza SSE-KMS con la característica opcional de claves de Bucket de S3, el valor del contexto de cifrado es el ARN del bucket.

```
"encryptionContext": {
    "aws:s3:arn": "arn:aws:s3:::amzn-s3-demo-bucket1"
}
```

Envío de solicitudes para objetos cifrados de AWS KMS

> **Important**
>
> Todas las solicitudes `GET` y `PUT` para los objetos cifrados de AWS KMS deben crearse mediante capa de conexión segura (SSL) o seguridad de la capa de transporte (TLS). Las solicitudes también deben firmarse con credenciales válidas, como AWS Signature Version 4 (o AWS Signature Version 2).

AWS Signature Version 4 es el proceso de agregar información de autenticación a las solicitudes de AWS enviadas por HTTP. Por seguridad, la mayoría de las solicitudes de AWS se firman con una clave de acceso, que se compone de un ID de clave de acceso y una clave de acceso secreta. Estas dos claves comúnmente se denominan credenciales de seguridad. Para obtener más información, consulte Autenticación de solicitudes (AWS Signature Version 4) y Proceso de firma de Signature Version 4.

> **Important**
>
> Si el objeto utiliza SSE-KMS, no envíe encabezados de solicitud de cifrado para solicitudes `GET` y `HEAD` . De lo contrario, aparecerá un error HTTP 400 Bad Request (Solicitud errónea).

Temas

- Especificación del cifrado del lado del servidor con AWS KMS (SSE-KMS)
- Reducción del costo de SSE-KMS con las claves de bucket de Amazon S3

Especificación del cifrado del lado del servidor con AWS KMS (SSE-KMS)

> **Important**
>
> Amazon S3 aplica ahora el cifrado del servidor con claves administradas por Amazon S3 (SSE-S3) como el nivel básico de cifrado para cada bucket de Amazon S3. Desde el 5 de

enero de 2023, todas las cargas de objetos nuevos a Amazon S3 se cifran automáticamente sin costo adicional y sin afectar al rendimiento. El estado de cifrado automático para la configuración de cifrado predeterminada en el bucket de S3 y para cargas de objetos nuevos está disponible en registros de AWS CloudTrail, Inventario de S3, Lente de almacenamiento de S3, la consola de Amazon S3 y como encabezado de respuesta a la API de Amazon S3 adicional en AWS Command Line Interface y los SDK de AWS. Para obtener más información, consulte Preguntas frecuentes del cifrado predeterminado.

Todos los buckets de Amazon S3 tienen el cifrado configurado de forma predeterminada y todos los objetos nuevos cargados en un bucket de S3 se cifran automáticamente en reposo. El cifrado del lado del servidor con claves administradas de Amazon S3 (SSE-S3) es la configuración de cifrado predeterminada para cada bucket de Amazon S3. Para usar otro tipo de cifrado, puede especificar el tipo de cifrado del servidor que se utilizará en las solicitudes PUT de S3 o puede establecer la configuración de cifrado predeterminada en el bucket de destino.

Si desea especificar un tipo de cifrado diferente en sus solicitudes PUT, puede utilizar el cifrado del servidor con claves de AWS Key Management Service (AWS KMS) (SSE-KMS), el cifrado del servidor de doble capa con claves de AWS KMS (DSSE-KMS) o el cifrado del servidor con claves proporcionadas por el cliente (SSE-C). Si desea establecer una configuración de cifrado predeterminada diferente en el bucket de destino, puede usar SSE-KMS o DSSE-KMS.

Puede aplicar cifrado cuando cargue un objeto nuevo o copie un objeto existente.

Puede especificar SSE-KMS mediante el uso de la consola de Amazon S3, las operaciones de la API de REST, los SDK de AWS y la AWS Command Line Interface (AWS CLI). Para obtener más información, consulte los siguientes temas.

Note

Puede utilizar AWS KMS keys de varias regiones en Amazon S3. No obstante, Amazon S3 trata las claves de varias regiones como si fueran claves de una sola región y no utiliza las características de varias regiones de la clave. Para obtener más información, consulte Uso de claves de varias regiones en la Guía para desarrolladores de AWS Key Management Service.

> **Note**
>
> Si quiere utilizar una clave de KMS propiedad de una cuenta diferente, primero debe tener permiso para utilizar la clave. Para obtener más información sobre los permisos entre cuentas para las claves de KMS, consulte Crear claves de KMS que otras cuentas puedan utilizar en la Guía para desarrolladores de AWS Key Management Service.

Uso de la consola de S3

En este tema se describe cómo configurar o cambiar el tipo de cifrado de un objeto para utilizar el cifrado del lado del servidor con claves de AWS Key Management Service (AWS KMS) (SSE-KMS) mediante la consola de Amazon S3.

> **Note**
>
> - Si cambia el cifrado de un objeto, se crea un nuevo objeto para reemplazar el antiguo. Si el control de versiones de S3 está activado, se crea una nueva versión del objeto y el objeto existente se convierte en una versión anterior. El rol que cambia la propiedad también se convierte en el propietario del nuevo objeto o (versión del objeto).
> - Si cambia el tipo de cifrado de un objeto que tiene etiquetas definidas por el usuario, debe tener el permiso `s3:GetObjectTagging`. Si va a cambiar el tipo de cifrado de un objeto que no tiene etiquetas definidas por el usuario pero que tiene un tamaño superior a 16 MB, también debe tener el permiso `s3:GetObjectTagging`.
>
>   Si la política de bucket de destino deniega la acción `s3:GetObjectTagging`, el tipo de cifrado del objeto se actualizará, pero las etiquetas definidas por el usuario se eliminarán del objeto y aparecerá un error.

Para añadir o cambiar el cifrado de un objeto

1. Inicie sesión en la AWS Management Console y abra la consola de Amazon S3 en https://console.aws.amazon.com/s3/.
2. En el panel de navegación izquierdo, elija Instancias.
3. En la lista Buckets (Buckets), elija el nombre del bucket que contiene el objeto.

4. En la lista Objects (Objetos), seleccione el nombre del objeto al que desea agregar cifrado o cuyo cifrado desea modificar.

   Aparece la página de detalles del objeto, con varias secciones que muestran las propiedades del objeto.

5. Elija la pestaña Propiedades.
6. Desplácese hacia abajo hasta la sección Configuración del cifrado del lado del servidor y elija Editar.

   Se abre la página Edit server-side encryption (Editar cifrado del lado del servidor).

7. En Cifrado del lado del servidor, en Configuración del cifrado, elija Anular la configuración predeterminada del bucket de cifrado.
8. En Tipo de cifrado, seleccione Cifrado del servidor con claves de AWS Key Management Service (SSE-KMS).

   > **Important**
   >
   > Si utiliza la opción de SSE-KMS para la configuración de cifrado predeterminado, se le aplicarán las cuotas de solicitudes por segundo (RPS) de AWS KMS. Para obtener más información acerca de las cuotas de AWS KMS y cómo solicitar un aumento de cuota, consulte Cuotas en la Guía para desarrolladores de AWS Key Management Service.

9. En Clave de AWS KMS, siga una de las siguientes opciones para elegir su clave de KMS:

   - Para seleccionar en una lista de claves de KMS disponibles, marque Elija entre sus claves de AWS KMS keys y seleccione su clave de KMS en la lista de claves disponibles.

     En esta lista aparecen tanto la Clave administrada de AWS (`aws/s3`) como las claves administradas por el cliente. Para obtener más información acerca de las claves administradas por el cliente, consulte Claves de cliente y claves de AWS en la Guía para desarrolladores de AWS Key Management Service.

   - Para introducir el ARN de la clave de KMS, elija Introducir el ARN de la clave de AWS KMS key e introduzca el ARN de la clave de KMS en el campo que aparece.
   - Para crear una nueva clave administrada por el cliente en la consola de AWS KMS, elija Crear una clave de KMS.

Para obtener más información acerca de cómo crear una AWS KMS key, consulte Creación de claves en la AWS Key Management Service Guía para desarrolladores.

> **Important**
>
> Solo puede utilizar las claves de KMS que estén disponibles en la misma Región de AWS del bucket. La consola de Amazon S3 solo muestra las primeras 100 claves de KMS de la misma región del bucket. Para utilizar una clave de KMS que no aparezca en la lista, debe introducir el ARN de la clave de KMS. Si desea utilizar una clave de KMS propiedad de una cuenta de diferente, primero debe tener permiso para utilizar la clave y, después, debe introducir el ARN de la clave de KMS.
> Amazon S3 admite solo claves KMS de cifrado simétricas y no claves KMS asimétricas. Para obtener más información, consulte Identificación de claves de KMS simétricas y asimétricas en la Guía para desarrolladores de AWS Key Management Service.

10. Elija Guardar cambios.

> **Note**
>
> En esta acción se aplica el cifrado a todos los objetos especificados. Al cifrar carpetas, espere a que finalice la operación de guardado para agregar nuevos objetos a la carpeta.

Uso de la API de REST

Cuando cree un objeto, es decir, cuando cargue un objeto nuevo o copie uno existente, podrá especificar el uso del cifrado del lado del servidor con AWS KMS keys (SSE-KMS) para cifrar los datos. Para ello, añada el encabezado `x-amz-server-side-encryption` a la solicitud. Configure el valor del encabezado para el algoritmo de cifrado `aws:kms`. Amazon S3 confirma que su objeto fue guardado con SSE-KMS al devolver el encabezado de respuesta `x-amz-server-side-encryption`.

Si especifica el encabezado `x-amz-server-side-encryption` con un valor de `aws:kms`, también puede utilizar los siguientes encabezados de solicitud:

- `x-amz-server-side-encryption-aws-kms-key-id`

- `x-amz-server-side-encryption-context`
- `x-amz-server-side-encryption-bucket-key-enabled`

Temas

- Las operaciones de la API de REST de Amazon S3 que admiten SSE-KMS
- Contexto de cifrado (x-amz-server-side-encryption-context)
- ID de clave de AWS KMS (x-amz-server-side-encryption-aws-kms-key-id)
- Claves de bucket de S3 (x-amz-server-side-encryption-aws-bucket-key-enabled)

Las operaciones de la API de REST de Amazon S3 que admiten SSE-KMS

Las siguientes operaciones de la API de REST aceptan los encabezados de solicitud `x-amz-server-side-encryption`, `x-amz-server-side-encryption-aws-kms-key-id` y `x-amz-server-side-encryption-context`.

- PutObject: cuando cargue datos mediante la operación de la API PUT, puede especificar estos encabezados de solicitud.
- CopyObject: al copiar un objeto, tiene un objeto de origen y otro de destino. Al pasar encabezados SSE-KMS con la operación `CopyObject`, estos se aplican solo al objeto de destino. Cuando copie un objeto existente, independientemente de si el objeto de origen está cifrado o no, el objeto de destino no estará cifrado, a no ser que solicite explícitamente el cifrado del lado en el servidor.
- POST Object: cuando utilice una operación POST para cargar un objeto, en vez de proporcionar los encabezados de solicitud, debe proporcionar la misma información en los campos del formulario.
- CreateMultipartUpload: cuando cargue objetos grandes mediante la operación de la API de carga multiparte, puede especificar estos encabezados. Especifique estos encabezados en la solicitud de `CreateMultipartUpload`.

Los encabezados de respuesta de las siguientes operaciones de la API de REST devuelven el encabezado `x-amz-server-side-encryption` cuando un objeto se almacena con el cifrado del lado del servidor.

- PutObject
- CopyObject
- POST Object

- CreateMultipartUpload
- UploadPart
- UploadPartCopy
- CompleteMultipartUpload
- GetObject
- HeadObject

> **Important**
>
> - Todas las solicitudes `GET` y `PUT` para un objeto protegido por AWS KMS producirán un error si no realiza dichas solicitudes con la capa de sockets seguros (SSL), la seguridad de la capa de transporte (TLS) o Signature Version 4.
> - Si su objeto utiliza SSE-KMS, no envíe encabezados de solicitud de cifrado para solicitudes `GET` y `HEAD` u obtendrá un error HTTP 400 BadRequest.

Contexto de cifrado (**`x-amz-server-side-encryption-context`**)

Si especifica `x-amz-server-side-encryption:aws:kms`, la API de Amazon S3 admitirá un contexto de cifrado con el encabezado `x-amz-server-side-encryption-context`. Un contexto de cifrado es un conjunto definido de pares clave-valor que contienen información contextual adicional sobre los datos.

Amazon S3 utiliza automáticamente el nombre de recurso de Amazon (ARN) del objeto o bucket como par del contexto de cifrado. Si utiliza SSE-KMS sin habilitar una clave de bucket de S3, utilice el ARN del objeto como contexto de cifrado, por ejemplo, `arn:aws:s3:::`*`object_ARN`*. Sin embargo, si utiliza SSE-KMS y habilita una clave de bucket de S3, utilice el ARN del bucket como contexto de cifrado, por ejemplo, `arn:aws:s3:::`*`bucket_ARN`*.

Si lo desea, puede proporcionar un par de contexto de cifrado adicional mediante el encabezado `x-amz-server-side-encryption-context`. No obstante, dado que el contexto de cifrado no está cifrado, asegúrese de no incluir información confidencial. Amazon S3 almacena este par de claves adicional junto con el contexto de cifrado predeterminado.

Para obtener información sobre el contexto de cifrado en Amazon S3, consulte Contexto de cifrado. Para obtener información general sobre el contexto de cifrado, consulte Conceptos de AWS Key

Management Service: contexto de cifrado en la Guía para desarrolladores de AWS Key Management Service.

ID de clave de AWS KMS (**`x-amz-server-side-encryption-aws-kms-key-id`**)

Puede utilizar el encabezado `x-amz-server-side-encryption-aws-kms-key-id` para especificar el ID de la clave administrada por el cliente utilizado para proteger los datos. Si especifica el encabezado `x-amz-server-side-encryption:aws:kms`, pero no proporciona el encabezado `x-amz-server-side-encryption-aws-kms-key-id`, Amazon S3 utiliza la Clave administrada de AWS (`aws/s3`) para proteger los datos. Si desea utilizar una clave administrada por el cliente, debe proporcionar el encabezado `x-amz-server-side-encryption-aws-kms-key-id` de dicha clave.

> **Important**
>
> Cuando utilice una AWS KMS key para el cifrado en el lado del servidor en Amazon S3, debe elegir una clave de cifrado de KMS simétrica. Amazon S3 solo admite claves KMS de cifrado simétricas. Para obtener más información sobre estas claves, consulte Symmetric encryption KMS keys (Claves de KMS de cifrado simétricas) en la Guía para desarrolladores de AWS Key Management Service.

Claves de bucket de S3 (**`x-amz-server-side-encryption-aws-bucket-key-enabled`**)

Puede utilizar el encabezado de solicitud `x-amz-server-side-encryption-aws-bucket-key-enabled` para habilitar o deshabilitar una clave de bucket de S3 en el nivel del objeto. Las claves de bucket de S3 reducen los costos de la solicitud de AWS KMS al disminuir el tráfico de solicitudes de Amazon S3 a AWS KMS. Para obtener más información, consulte Reducción del costo de SSE-KMS con las claves de bucket de Amazon S3.

Si especifica el encabezado `x-amz-server-side-encryption:aws:kms`, pero no proporciona el encabezado `x-amz-server-side-encryption-aws-bucket-key-enabled`, el objeto utiliza el ajuste de la clave de bucket de S3 para que el bucket de destino cifre el objeto. Para obtener más información, consulte Configuración de una clave de bucket de S3 en el nivel de objeto .

Mediante AWS CLI

Para utilizar los siguientes comandos de ejemplo de la AWS CLI, sustituya *`user input placeholders`* con su información.

Cuando se carga un objeto nuevo o se copia uno existente, puede especificar el uso del cifrado del lado del servidor con claves de AWS KMS para cifrar los datos. Para ello, añada el encabezado `--server-side-encryption aws:kms` a la solicitud. Utilice `--ssekms-key-id` *`example-key-id`* para agregar la clave de AWS KMS administrada de cliente que ha creado. Si especifica `--server-side-encryption aws:kms`, pero no proporciona un ID de clave de AWS KMS, Amazon S3 utilizará una clave administrada de AWS.

```
aws s3api put-object --bucket amzn-s3-demo-bucket --key example-object-key --
server-side-encryption aws:kms --ssekms-key-id example-key-id --ssekms-encryption-
context example-encryption-context --body filepath
```

Puede habilitar o deshabilitar las claves de bucket de S3 en las operaciones `put-object` o `copy-object` agregando `--bucket-key-enabled` o `--no-bucket-key-enabled`. Las claves de bucket de S3 pueden reducir los costos de solicitud de AWS KMS al disminuir el tráfico de solicitudes de Amazon S3 a AWS KMS. Para obtener más información, consulte Reducción del costo de SSE-KMS con las claves de bucket de S3.

```
aws s3api put-object --bucket amzn-s3-demo-bucket --key example-object-key --server-
side-encryption aws:kms --bucket-key-enabled --body filepath
```

Puede copiar un objeto de un bucket de origen a uno nuevo y especificar el cifrado SSE-KMS.

```
aws s3api copy-object --copy-source amzn-s3-demo-bucket/example-object-key --
bucket amzn-s3-demo-bucket2 --key example-object-key --server-side-encryption aws:kms
 --sse-kms-key-id example-key-id --ssekms-encryption-context example-encryption-context
```

Uso de los AWS SDK

Al utilizar los SDK de AWS, puede solicitar a Amazon S3 que utilice AWS KMS keys para el cifrado del lado del servidor. En los ejemplos siguientes se muestra cómo usar SSE-KMS con los AWS SDK para Java y .NET. Para obtener información acerca de otros SDK, consulte Código de muestra y bibliotecas en el Centro de desarrolladores de AWS.

> **Important**
>
> Cuando utilice una AWS KMS key para el cifrado en el lado del servidor en Amazon S3, debe elegir una clave de cifrado de KMS simétrica. Amazon S3 solo admite claves KMS de cifrado simétricas. Para obtener más información sobre estas claves, consulte Symmetric encryption

KMS keys (Claves de KMS de cifrado simétricas) en la Guía para desarrolladores de AWS Key Management Service.

Operación de **`CopyObject`**

Si copia objetos, agrega las mismas propiedades de la solicitud (`ServerSideEncryptionMethod` y `ServerSideEncryptionKeyManagementServiceKeyId`) para solicitar a Amazon S3 que use una AWS KMS key. Para obtener más información acerca de la copia de objetos, consulte Copia, traslado y cambio de nombre de objetos.

Operación de **`PUT`**

Java

Si carga un objeto mediante el AWS SDK for Java, puede solicitar que Amazon S3 utilice una AWS KMS key si agrega la propiedad `SSEAwsKeyManagementParams`, tal y como se muestra en la siguiente solicitud:

```
PutObjectRequest putRequest = new PutObjectRequest(bucketName,
   keyName, file).withSSEAwsKeyManagementParams(new SSEAwsKeyManagementParams());
```

En este caso, Amazon S3 utiliza la Clave administrada de AWS (`aws/s3`). Para obtener más información, consulte Uso del cifrado del servidor con claves de AWS KMS (SSE-KMS). Opcionalmente, puede crear una clave KMS de cifrado simétrica y especificarla en la solicitud, tal y como se muestra en el siguiente ejemplo:

```
PutObjectRequest putRequest = new PutObjectRequest(bucketName,
   keyName, file).withSSEAwsKeyManagementParams(new
 SSEAwsKeyManagementParams(keyID));
```

Para obtener más información acerca de la creación de claves administradas por el cliente, consulte Programación de la API de AWS KMS en la Guía para desarrolladores de AWS Key Management Service.

Para ver ejemplos de código funcional sobre cómo cargar un objeto, consulte los temas siguientes. Para utilizar estos ejemplos, tiene que actualizar estos ejemplos de código y facilitar la información de cifrado, tal como se muestra en el fragmento de código anterior.

- Para cargar un objeto en una única operación, consulte Carga de objetos.
- Para las cargas multiparte que utilizan las operaciones de API de carga multiparte de nivel alto o de nivel bajo, consulte Carga de un objeto con la carga multiparte.

.NET

Si carga un objeto mediante el AWS SDK for .NET, puede solicitar que Amazon S3 utilice una AWS KMS key si agrega la propiedad `ServerSideEncryptionMethod`, tal y como se muestra en la siguiente solicitud:

```
PutObjectRequest putRequest = new PutObjectRequest
 {
     BucketName = amzn-s3-demo-bucket,
     Key = keyName,
     // other properties
     ServerSideEncryptionMethod = ServerSideEncryptionMethod.AWSKMS
 };
```

En este caso, Amazon S3 utiliza la Clave administrada de AWS. Para obtener más información, consulte Uso del cifrado del servidor con claves de AWS KMS (SSE-KMS). Opcionalmente, puede crear su propia clave administrada por el cliente de cifrado simétrica y especificarla en la solicitud, tal y como se muestra en el siguiente ejemplo:

```
PutObjectRequest putRequest1 = new PutObjectRequest
{
  BucketName = amzn-s3-demo-bucket,
  Key = keyName,
  // other properties
  ServerSideEncryptionMethod = ServerSideEncryptionMethod.AWSKMS,
  ServerSideEncryptionKeyManagementServiceKeyId = keyId
};
```

Para obtener más información acerca de la creación de claves administradas por el cliente, consulte Programación de la API de AWS KMS en la Guía para desarrolladores de AWS Key Management Service.

Para ver ejemplos de código funcional sobre cómo cargar un objeto, consulte los temas siguientes. Para utilizar estos ejemplos, tiene que actualizar estos ejemplos de código y facilitar la información de cifrado, tal como se muestra en el fragmento de código anterior.

- Para cargar un objeto en una única operación, consulte Carga de objetos.
- Para las cargas multiparte que utilizan las operaciones de API de carga multiparte de nivel alto o de nivel bajo, consulte Carga de un objeto con la carga multiparte.

URL prefirmadas

Java

Si crea una dirección URL prefirmada para un objeto cifrado con una AWS KMS key, es necesario que se especifique explícitamente Signature Version 4, tal y como se muestra en el siguiente ejemplo:

```
ClientConfiguration clientConfiguration = new ClientConfiguration();
clientConfiguration.setSignerOverride("AWSS3V4SignerType");
AmazonS3Client s3client = new AmazonS3Client(
        new ProfileCredentialsProvider(), clientConfiguration);
...
```

Para ver un ejemplo del código, consulte Uso compartido de objetos con URL prefirmadas.

.NET

Si crea una dirección URL prefirmada para un objeto cifrado con una AWS KMS key, es necesario que se especifique explícitamente Signature Version 4, tal y como se muestra en el siguiente ejemplo:

```
AWSConfigs.S3Config.UseSignatureVersion4 = true;
```

Para ver un ejemplo del código, consulte Uso compartido de objetos con URL prefirmadas.

Reducción del costo de SSE-KMS con las claves de bucket de Amazon S3

Las claves de bucket de Amazon S3 reducen el costo del cifrado del lado del servidor de Amazon S3 con las claves de AWS Key Management Service (AWS KMS) (SSE-KMS). Usar una clave de nivel de bucket para SSE-KMS puede reducir los costos de la solicitud de AWS KMS hasta en un 99 %, ya que disminuye el tráfico de solicitudes de Amazon S3 a AWS KMS. Con unos pocos clics en la AWS Management Console y sin modificar sus aplicaciones de cliente, puede configurar su bucket de modo que utilice una clave de bucket de S3 para el cifrado SSE-KMS en los objetos nuevos.

> **Note**
>
> Las claves de bucket de S3 no son compatibles con el cifrado de doble capa del servidor con claves de AWS Key Management Service (AWS KMS) (DSSE-KMS).

Claves de bucket de S3 para SSE-KMS

Las cargas de trabajo que acceden a millones o miles de millones de objetos cifrados con SSE-KMS pueden generar grandes volúmenes de solicitudes para AWS KMS. Cuando utiliza SSE-KMS para proteger los datos sin una clave de bucket de S3, Amazon S3 recurre a AWS KMS para utilizar una clave de datos individual por cada objeto. En ese caso, Amazon S3 realiza una llamada a AWS KMS cada vez que se realiza una solicitud respecto de un objeto con cifrado de KMS. Para obtener información sobre cómo funciona SSE-KMS, consulte Uso del cifrado del servidor con claves de AWS KMS (SSE-KMS).

Cuando configura el bucket para utilizar una clave de bucket de S3 para SSE-KMS, AWS genera una clave en el nivel de bucket de corta duración desde AWS KMS y, a continuación, la mantiene temporalmente en S3. Esta clave en el nivel de bucket creará claves de datos para objetos nuevos durante el ciclo de vida. Las claves de bucket de S3 se utilizan durante un periodo de tiempo limitado dentro de Amazon S3, lo que reduce la necesidad de que S3 realice solicitudes a AWS KMS para completar las operaciones de cifrado. De esta manera, se reduce el tráfico de S3 a AWS KMS, lo que le permite acceder a objetos cifrados con AWS KMS en Amazon S3 a una fracción del costo anterior.

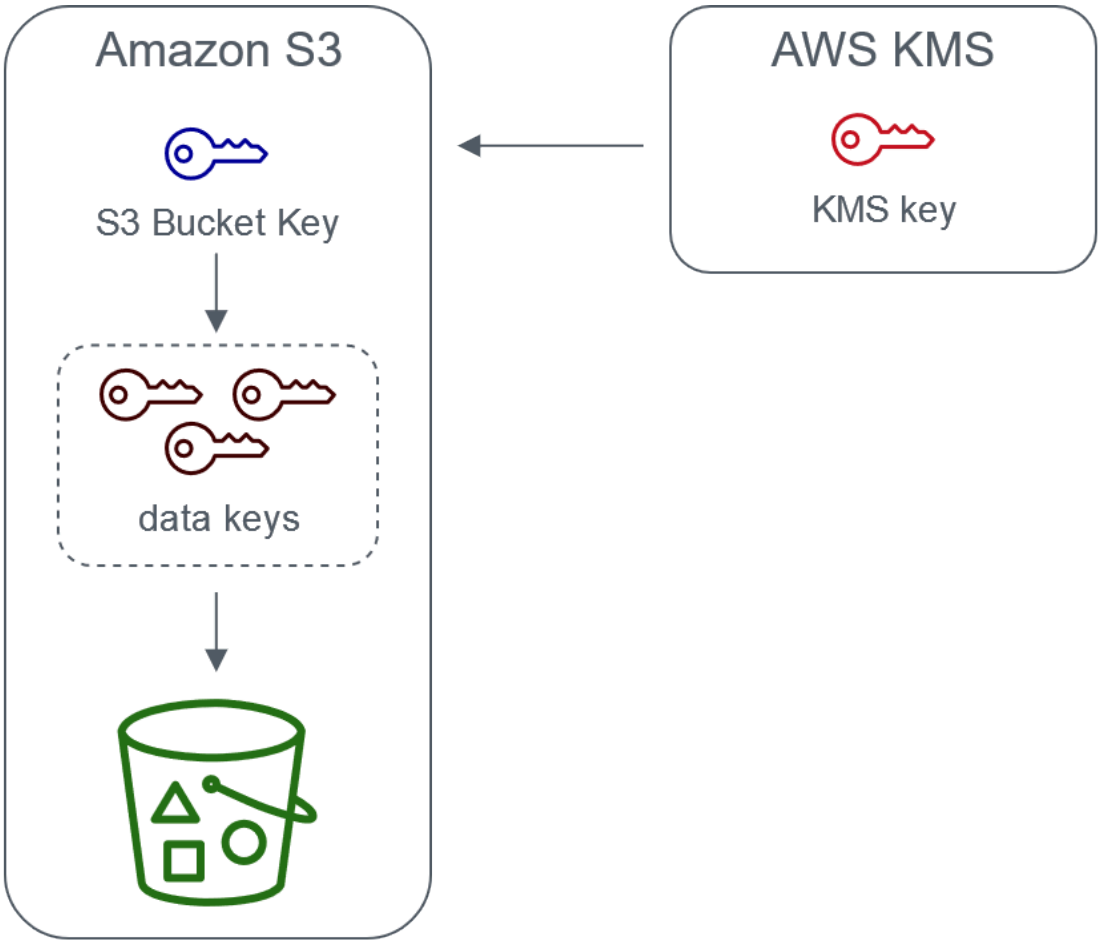
Las claves únicas en el nivel de bucket se obtienen al menos una vez por solicitante para garantizar que el acceso del solicitante a la clave se capture en un evento de AWS KMS CloudTrail. Amazon S3 trata a los intermediarios como solicitantes diferentes cuando utilizan roles o cuentas diferentes, o el mismo rol con distintas políticas de alcance. Los ahorros en las solicitudes de AWS KMS reflejan el número de solicitantes, los patrones de solicitudes y la antigüedad relativa de los objetos solicitados. Por ejemplo, reducir el número de solicitantes, solicitar varios objetos en un periodo de tiempo limitado y cifrarlos con la misma clave de nivel de bucket sigue generando mayores ahorros.

> **Note**
>
> El uso de claves de bucket de S3 le permite ahorrar costos de solicitud de AWS KMS al reducir las solicitudes a AWS KMS para operaciones `Encrypt`, `GenerateDataKey` y `Decrypt` mediante el uso de una clave de nivel de bucket. Por diseño, las solicitudes

posteriores que aprovechan esta clave de nivel de bucket no generan solicitudes de API de AWS KMS ni validan el acceso con respecto a la política de claves de AWS KMS.

Cuando se configura una clave de bucket de S3, los objetos que ya están en el bucket no utiliza la clave de bucket de S3. Para configurar una clave de bucket de S3 para los objetos existentes, puede utilizar una operación `CopyObject`. Para obtener más información, consulte Configuración de una clave de bucket de S3 en el nivel de objeto .

Amazon S3 solo compartirá una clave de Bucket de S3 para los objetos que se cifren con la misma AWS KMS key. Las claves de bucket de S3 son compatibles con las claves de KMS creadas por AWS KMS, el material de claves importado y el material de claves respaldado por almacenes de claves personalizados.



Server-side encryption with AWS Key Management service using an S3 Bucket Key

Configuración de claves de bucket de S3

Es posible configurar su bucket para utilizar una clave de bucket de S3 para SSE-KMS en objetos nuevos a través de la consola de Amazon S3, los SDK de AWS, la AWS CLI o la API de REST. Con las claves de bucket de S3 habilitadas en el bucket, los objetos cargados con una clave de SSE-KMS especificada diferente utilizarán sus propias claves de bucket de S3. Independientemente de la configuración de la clave de bucket de S3, puede incluir el encabezado `x-amz-server-side-encryption-bucket-key-enabled` con un valor `true` o `false` en la solicitud para invalidar la configuración del bucket.

Antes de configurar el bucket para utilizar una clave de bucket de S3, consulte Cambios para tener en cuenta antes de habilitar una clave de bucket de S3.

Configuración de una clave de bucket de S3 mediante la consola de Amazon S3

Al crear un nuevo bucket, puede configurarlo para utilizar una clave de bucket de S3 para SSE-KMS en objetos nuevos. También puede configurar un bucket existente para utilizar una clave de bucket de S3 para SSE-KMS en objetos nuevos al actualizar las propiedades del bucket.

Para obtener más información, consulte Configuración del bucket para utilizar una clave de bucket de S3 con SSE-KMS para objetos nuevos.

Compatibilidad con la API de REST, la AWS CLI y el SDK de AWS para claves de buckets de S3

Puede emplear la API de REST, la AWS CLI o el SDK de AWS a fin de configurar el bucket de manera que utilice una clave de bucket de S3 para SSE-KMS en los objetos nuevos. También puede habilitar una clave de bucket de S3 en el nivel de objeto.

Para más información, consulte los siguientes temas:

- Configuración de una clave de bucket de S3 en el nivel de objeto
- Configuración del bucket para utilizar una clave de bucket de S3 con SSE-KMS para objetos nuevos

Las siguientes operaciones de la API admiten claves de bucket de S3 para SSE-KMS:

- PutBucketEncryption
  - `ServerSideEncryptionRule` acepta el `BucketKeyEnabled` parámetro para habilitar y deshabilitar una clave de bucket de S3.

- GetBucketEncryption
  - `ServerSideEncryptionRule` devuelve la configuración de `BucketKeyEnabled`.
- PutObject, CopyObject, CreateMutlipartUpload y PostObject
  - El encabezado de solicitud `x-amz-server-side-encryption-bucket-key-enabled` habilita o deshabilita una clave de bucket de S3 en el nivel de objeto.
- HeadObject, GetObject, UploadPartCopy, UploadPart y CompleteMultipartUpload
  - El encabezado de respuesta `x-amz-server-side-encryption-bucket-key-enabled` indica si una clave de bucket de S3 está habilitada o deshabilitada para un objeto.

Uso de AWS CloudFormation

En AWS CloudFormation, el recurso `AWS::S3::Bucket` contiene una propiedad de cifrado denominada `BucketKeyEnabled` que usted puede utilizar para activar o desactivar una clave de bucket de S3.

Para obtener más información, consulte Uso de AWS CloudFormation.

Cambios para tener en cuenta antes de habilitar una clave de bucket de S3

Antes de habilitar una clave de bucket de S3, tenga en cuenta los siguientes cambios relacionados:

Políticas de IAM o de claves de AWS KMS

Si sus políticas de AWS Identity and Access Management (IAM) o sus políticas de clave de AWS KMS existentes utilizan el nombre de recurso de Amazon (ARN) del objeto como contexto de cifrado para ajustar o limitar el acceso a su clave de KMS, estas políticas no funcionarán con una clave de bucket de S3. Las claves de bucket de S3 utilizan el ARN del bucket como contexto de cifrado. Antes de habilitar una clave de bucket de S3, actualice las políticas de IAM o las políticas de clave de AWS KMS de manera que utilicen el ARN del bucket como contexto de cifrado.

Para obtener más información sobre el contexto de cifrado y las claves de bucket de S3, consulte Contexto de cifrado.

Eventos de CloudTrail para AWS KMS

Después de habilitar una clave de bucket de S3, los eventos de AWS KMS CloudTrail registran el ARN del bucket en lugar del ARN del objeto. Además, en sus registros verá menos eventos de KMS CloudTrail para objetos SSE-KMS. Dado que el material clave tiene un tiempo limitado en Amazon S3, se realizan menos solicitudes a AWS KMS.

Uso de una clave de bucket de S3 con replicación

Puede usar claves de bucket de S3 con replicación en la misma región (SRR) y replicación entre regiones (CRR).

Cuando Amazon S3 replica un objeto cifrado, generalmente conserva la configuración de cifrado del objeto replicado en el bucket de destino. Sin embargo, si el objeto de origen no está cifrado y el bucket de destino utiliza el cifrado predeterminado o una clave de bucket de S3, Amazon S3 cifra el objeto con la configuración del bucket de destino.

Los siguientes ejemplos ilustran cómo funciona una clave de bucket de S3 con la replicación. Para obtener más información, consulte Replicación de objetos cifrados (SSE-C, SSE-S3, SSE-KMS, DSSE-KMS).

Example Ejemplo 1: el objeto de origen utiliza claves de bucket de S3; el bucket de destino utiliza el cifrado predeterminado

Si el objeto de origen utiliza una clave de bucket de S3 pero el bucket de destino utiliza el cifrado predeterminado con SSE-KMS, el objeto de réplica mantiene su configuración de cifrado de clave de bucket de S3 en el bucket de destino. El bucket de destino sigue utilizando el cifrado predeterminado con SSE-KMS.

Example Ejemplo 2: el objeto de origen no está cifrado; el bucket de destino utiliza una clave de bucket de S3 con SSE-KMS

Si el objeto de origen no está cifrado y el bucket de destino utiliza una clave de bucket de S3 con SSE-KMS, el objeto de réplica se cifra con una clave de bucket de S3 que utiliza SSE-KMS en el bucket de destino. Esto hace que el elemento `ETag` del objeto de origen sea diferente al elemento `ETag` del objeto de réplica. Debe actualizar las aplicaciones que utilicen el elemento `ETag` para incluir esta diferencia.

Trabajar con claves de bucket de S3

Para obtener más información sobre cómo habilitar claves de bucket de S3 y trabajar con ellas, consulte las siguientes secciones:

- Configuración del bucket para utilizar una clave de bucket de S3 con SSE-KMS para objetos nuevos
- Configuración de una clave de bucket de S3 en el nivel de objeto
- Visualización de la configuración de una clave de bucket de S3

Configuración del bucket para utilizar una clave de bucket de S3 con SSE-KMS para objetos nuevos

Cuando configura el cifrado del lado del servidor con claves de AWS Key Management Service (AWS KMS) (SSE-KMS), puede configurar el bucket para utilizar una clave de bucket de S3 para SSE-KMS en los objetos nuevos. Las claves de bucket de S3 reducen el tráfico de solicitudes de Amazon S3 a AWS KMS, así como el costo de SSE-KMS. Para obtener más información, consulte Reducción del costo de SSE-KMS con las claves de bucket de Amazon S3.

Puede configurar su bucket de manera que utilice una clave de bucket de S3 para SSE-KMS en los objetos nuevos mediante la consola de Amazon S3, la API de REST, los SDK de AWS, la AWS Command Line Interface (AWS CLI) o AWS CloudFormation. Si desea habilitar o deshabilitar una clave de bucket de S3 para los objetos existentes, puede utilizar una operación `CopyObject`. Para obtener más información, consulte Configuración de una clave de bucket de S3 en el nivel de objeto y Uso de la herramienta de operaciones por lotes de S3 para cifrar objetos con claves de bucket de S3.

Cuando se habilita una clave de bucket de S3 para el bucket de origen o de destino, el contexto de cifrado será el nombre de recurso de Amazon (ARN) del bucket y no el ARN del objeto, por ejemplo, `arn:aws:s3:::`*`bucket_ARN`*. Debe actualizar las políticas de IAM para utilizar el ARN del bucket para el contexto de cifrado. Para obtener más información, consulte Claves de bucket y replicación de S3.

Los siguientes ejemplos ilustran cómo funciona una clave de bucket de S3 con la replicación. Para obtener más información, consulte Replicación de objetos cifrados (SSE-C, SSE-S3, SSE-KMS, DSSE-KMS).

Requisitos previos

Antes de configurar el bucket para utilizar una clave de bucket de S3, consulte Cambios para tener en cuenta antes de habilitar una clave de bucket de S3.

Uso de la consola de S3

En la consola de S3, puede habilitar o deshabilitar una clave de bucket de S3 para un bucket nuevo o uno existente. Los objetos en la consola de S3 hereda el ajuste de la clave de bucket de S3 de la configuración del bucket. Cuando habilita una clave de bucket de S3 para su bucket, los nuevos objetos que carga en el bucket utilizan una clave de bucket de S3 para SSE-KMS.

Cargar, copiar o modificar objetos en buckets que tienen habilitada una clave de bucket de S3

Si carga, modifica o copia un objeto en un bucket que tiene habilitada una clave de bucket de S3, el ajuste de la clave de bucket de S3 para ese objeto podría actualizarse para alinearse con la configuración del bucket.

Si un objeto ya tiene habilitada una clave de bucket de S3, el ajuste de la clave de bucket de S3 para ese objeto no cambia al copiar o modificar el objeto. Sin embargo, si modifica o copia un objeto que no tiene habilitada una clave de bucket de S3 y el bucket de destino tiene una configuración de clave de bucket de S3, el objeto hereda el ajuste de la clave de bucket de S3 del bucket de destino. Por ejemplo, si el objeto de origen no tiene habilitada una clave de bucket de S3 pero el bucket de destino tiene habilitada la clave de bucket de S3, se habilita una clave de bucket de S3 para el objeto.

Para habilitar una clave de bucket de S3 al crear un nuevo bucket,

1. Inicie sesión en la AWS Management Console y abra la consola de Amazon S3 en https://console.aws.amazon.com/s3/.
2. En el panel de navegación izquierdo, elija Instancias.
3. Elija Crear bucket.
4. Ingrese el nombre del bucket y elija la Región de AWS.
5. En Cifrado predeterminado, en Tipo de clave de cifrado, elija Clave de AWS Key Management Service (SSE-KMS).
6. En Clave de AWS KMS, siga una de las siguientes opciones para elegir su clave de KMS:

   - Para seleccionar en una lista de claves de KMS disponibles, marque Elija entre sus claves de AWS KMS keys y seleccione su clave de KMS en la lista de claves disponibles.

     En esta lista aparecen tanto la Clave administrada de AWS (`aws/s3`) como las claves administradas por el cliente. Para obtener más información acerca de las claves administradas por el cliente, consulte Claves de cliente y claves de AWS en la Guía para desarrolladores de AWS Key Management Service.
   - Para introducir el ARN de la clave de KMS, elija Introducir el ARN de la AWS KMS key e introduzca el ARN de la clave de KMS en el campo que aparece.
   - Para crear una nueva clave administrada por el cliente en la consola de AWS KMS, elija Crear una clave de KMS.

     Para obtener más información acerca de cómo crear una AWS KMS key, consulte Creación de claves en la AWS Key Management Service Guía para desarrolladores.

7. En Bucket Key (Clave de bucket), seleccione Enable (Habilitar).
8. Elija Create bucket (Crear bucket).

   Amazon S3 crea el bucket con una clave de bucket de S3 habilitada. Los nuevos objetos que cargue en el bucket utilizarán una clave de bucket de S3.

   Para deshabilitar una clave de bucket de S3, siga los pasos anteriores y elija Disable (Deshabilitar).

Para habilitar una clave de bucket de S3 para un bucket existente,

1. Abra la consola de Amazon S3 en https://console.aws.amazon.com/s3.
2. En el panel de navegación situado a la izquierda, elija Buckets.
3. En la lista Buckets, elija el bucket para el que desea habilitar una clave de bucket de S3.
4. Elija la pestaña Propiedades.
5. En Cifrado predeterminado, elija Editar.
6. En Cifrado predeterminado, en Tipo de clave de cifrado, elija Clave de AWS Key Management Service (SSE-KMS).
7. En Clave de AWS KMS, siga una de las siguientes opciones para elegir su clave de KMS:

   - Para seleccionar en una lista de claves de KMS disponibles, marque Elija entre sus claves de AWS KMS keys y seleccione su clave de KMS en la lista de claves disponibles.

     En esta lista aparecen tanto la Clave administrada de AWS (`aws/s3`) como las claves administradas por el cliente. Para obtener más información acerca de las claves administradas por el cliente, consulte Claves de cliente y claves de AWS en la Guía para desarrolladores de AWS Key Management Service.
   - Para introducir el ARN de la clave de KMS, elija Introducir el ARN de la AWS KMS key e introduzca el ARN de la clave de KMS en el campo que aparece.
   - Para crear una nueva clave administrada por el cliente en la consola de AWS KMS, elija Crear una clave de KMS.

     Para obtener más información acerca de cómo crear una AWS KMS key, consulte Creación de claves en la AWS Key Management Service Guía para desarrolladores.
8. En Bucket Key (Clave de bucket), seleccione Enable (Habilitar).
9. Elija Save changes.

Amazon S3 habilita una clave de bucket de S3 para los nuevos objetos agregados al bucket. Los objetos existentes no utilizan la clave de bucket de S3. Para configurar una clave de bucket de S3 para los objetos existentes, puede utilizar una operación `CopyObject`. Para obtener más información, consulte Configuración de una clave de bucket de S3 en el nivel de objeto .

Para deshabilitar una clave de bucket de S3, siga los pasos anteriores y elija Disable (Deshabilitar).

Uso de la API de REST

Puede utilizar PutBucketEncryption para habilitar o desactivar una clave de bucket de S3 en su bucket. Para configurar una clave de bucket de S3 con `PutBucketEncryption`, especifique el tipo de dato ServerSideEncryptionRule, que incluye el cifrado predeterminado con SSE-KMS. También puede utilizar una clave administrada por el cliente al especificar el ID de clave de KMS para la clave administrada por el cliente.

Para obtener más información y sintaxis de ejemplo, consulte PutBucketEncryption.

Uso de AWS SDK para Java

En el siguiente ejemplo, se habilita el cifrado de bucket predeterminado con SSE-KMS y una clave de bucket de S3 mediante 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);
```

Uso de la AWS CLI

En el siguiente ejemplo, se habilita el cifrado de bucket predeterminado con SSE-KMS y una clave de bucket de S3 mediante AWS CLI. Reemplace los *user input placeholders* con su propia información.

```
aws s3api put-bucket-encryption --bucket amzn-s3-demo-bucket --server-side-encryption-
configuration '{
        "Rules": [
            {
                "ApplyServerSideEncryptionByDefault": {
                    "SSEAlgorithm": "aws:kms",
                    "KMSMasterKeyID": "KMS-Key-ARN"
                },
                "BucketKeyEnabled": true
            }
        ]
    }'
```

Uso de AWS CloudFormation

Para obtener más información sobre cómo configurar una clave de bucket de S3 con AWS CloudFormation, consulte AWS::S3::Bucket ServerSideEncryptionRule en la Guía del usuario de AWS CloudFormation.

Configuración de una clave de bucket de S3 en el nivel de objeto

Cuando realiza una operación PUT o COPY mediante la API de REST, los SDK de AWS o la AWS CLI, puede habilitar o desactivar una clave de bucket de S3 en el nivel de objeto agregando el encabezado de la solicitud `x-amz-server-side-encryption-bucket-key-enabled` con un valor `true` o `false`. Las claves de bucket de S3 permiten reducir el costo de cifrado del lado del servidor mediante AWS Key Management Service (AWS KMS) (SSE-KMS) al disminuir el tráfico de solicitudes de Amazon S3 a AWS KMS. Para obtener más información, consulte Reducción del costo de SSE-KMS con las claves de bucket de Amazon S3.

Cuando se configura una clave de bucket de S3 para un objeto mediante una operación PUT o COPY, Amazon S3 solo actualiza la configuración de ese objeto. El ajuste de la clave de bucket

de S3 para el bucket de destino no cambia. Si envía una solicitud PUT o COPY para un objeto cifrado con KMS a un bucket con las claves de bucket de S3 habilitadas, la operación en el nivel de objeto utilizará automáticamente las claves de bucket de S3, a menos que desactive las claves del encabezado de la solicitud. Si no especifica una clave de bucket de S3 para el objeto, Amazon S3 aplica el ajuste de la clave de bucket de S3 para el bucket de destino al objeto.

Requisito previo:

Antes de configurar el objeto para usar una clave de bucket de S3, consulte Cambios para tener en cuenta antes de habilitar una clave de bucket de S3.

Temas

- Herramienta de operaciones por lotes de Amazon S3
- Uso de la API de REST
- Uso de AWS SDK para Java (PutObject)
- Uso de la AWS CLI (PutObject)

Herramienta de operaciones por lotes de Amazon S3

Para cifrar los objetos de Amazon S3 existentes, puede utilizar la herramienta de operaciones por lotes de Amazon S3. A las operaciones por lotes de S3 se les proporciona una lista de objetos en los que deben actuar. Las operaciones por lotes llaman a la API correspondiente para llevar a cabo la operación especificada.

Puede utilizar la operación de copia de la herramienta de operaciones por lotes de S3 para copiar objetos existentes sin cifrar y escribirlos como objetos cifrados en el mismo bucket. Un solo trabajo de la herramienta de operaciones por lotes puede realizar la operación especificada en miles de millones de objetos. Para obtener más información, consulte Realización de operaciones por lotes a gran escala en objetos de Amazon S3 y Cifrado de objetos con la herramienta de operaciones por lotes de Amazon S3.

Uso de la API de REST

Si utiliza SSE-KMS, puede habilitar una clave de bucket de S3 para un objeto con las siguientes operaciones de la API:

- PutObject : al cargar un objeto, puede especificar el encabezado de solicitud `x-amz-server-side-encryption-bucket-key-enabled` para habilitar o deshabilitar una clave de bucket de S3 en el nivel de objeto.

- CopyObject : al copiar un objeto y configurar SSE-KMS, puede especificar el encabezado de solicitud `x-amz-server-side-encryption-bucket-key-enabled` para habilitar o deshabilitar una clave de bucket de S3 para el objeto.
- POST Object: al utilizar una operación `POST` para cargar un objeto y configurar SSE-KMS, puede utilizar el campo de formulario `x-amz-server-side-encryption-bucket-key-enabled` para habilitar o deshabilitar una clave de bucket de S3 para el objeto.
- CreateMultipartUpload: al cargar objetos grandes mediante la operación de la API `CreateMultipartUpload` y configurar SSE-KMS, puede usar el encabezado de solicitud `x-amz-server-side-encryption-bucket-key-enabled` para habilitar o deshabilitar una clave de bucket de S3 para el objeto.

Para habilitar una clave de bucket de S3 en el nivel de objeto, incluya el encabezado de solicitud `x-amz-server-side-encryption-bucket-key-enabled`. Para obtener más información sobre SSE-KMS y la API de REST, consulte Uso de la API de REST.

Uso de AWS SDK para Java (PutObject)

Puede utilizar el siguiente ejemplo para configurar una clave de bucket de S3 en el nivel de objeto mediante AWS SDK for Java.

Java

```
AmazonS3 s3client = AmazonS3ClientBuilder.standard()
    .withRegion(Regions.DEFAULT_REGION)
    .build();

String bucketName = "amzn-s3-demo-bucket1";
String keyName = "key name for object";
String contents = "file contents";

PutObjectRequest putObjectRequest = new PutObjectRequest(bucketName, keyName,
 contents)
    .withBucketKeyEnabled(true);

s3client.putObject(putObjectRequest);
```

Uso de la AWS CLI (PutObject)

Puede utilizar el siguiente ejemplo de la AWS CLI para configurar una clave de bucket de S3 en el nivel de objeto como parte de una solicitud `PutObject`.

```
aws s3api put-object --bucket amzn-s3-demo-bucket --key object key name --server-side-
encryption aws:kms --bucket-key-enabled --body filepath
```

Visualización de la configuración de una clave de bucket de S3

Puede visualizar la configuración de una clave de bucket de S3 en el nivel de bucket o de objeto mediante la consola de Amazon S3, la API de REST, la AWS Command Line Interface (AWS CLI) o los SDK de AWS.

Las claves de bucket de S3 permiten reducir el tráfico de solicitudes de Amazon S3 a AWS KMS así como el costo del cifrado del lado del servidor mediante AWS Key Management Service (SSE-KMS). Para obtener más información, consulte Reducción del costo de SSE-KMS con las claves de bucket de Amazon S3.

Para ver el ajuste de la clave de bucket de S3 para un bucket o un objeto que ha heredado el ajuste de la clave de bucket de S3 de la configuración del bucket, necesita permiso para realizar la acción `s3:GetEncryptionConfiguration`. Para obtener más información, consulte GetBucketEncryption en la Referencia de la API de Amazon Simple Storage Service.

Uso de la consola de S3

En la consola de S3, puede ver el ajuste de la clave de bucket de S3 para el bucket u objeto. El ajuste de la clave de bucket de S3 se hereda de la configuración del bucket a menos que los objetos de origen ya tengan configurada una clave de bucket de S3.

Los objetos y las carpetas del mismo bucket pueden tener diferentes ajustes de clave de bucket de S3. Por ejemplo, si carga un objeto con la API de REST y habilita una clave de bucket de S3 para el objeto, este último conservará el ajuste de la clave de bucket de S3 en el bucket de destino, aun cuando la clave de bucket de S3 esté deshabilitada en el bucket de destino. Como otro ejemplo, si habilita una clave de bucket de S3 para un bucket existente, los objetos que ya estén en el bucket no utilizan una clave de bucket de S3. Sin embargo, los objetos nuevos tienen habilitada una clave de bucket de S3.

Para ver el ajuste de la clave de bucket de S3 para su bucket

1. Inicie sesión AWS Management Console Management Console y abra la consola de Amazon S3 en https://console.aws.amazon.com/s3/.
2. En el panel de navegación izquierdo, elija Instancias.
3. En la lista Buckets , elija el bucket para el que desea habilitar una clave de bucket de S3.
4. Seleccione Properties (Propiedades).
5. En la sección Default encryption (Cifrado predeterminado), en Bucket Key (Clave de bucket), verá el ajuste de la clave de bucket de S3 para su bucket.

   Si no puede verlo, es posible que no tenga permiso para realizar la acción `s3:GetEncryptionConfiguration`. Para obtener más información, consulte GetBucketEncryption en la Referencia de la API de Amazon Simple Storage Service.

Para ver el ajuste de la clave de bucket de S3 para su objeto,

1. Inicie sesión en la AWS Management Console y abra la consola de Amazon S3 en https://console.aws.amazon.com/s3/.
2. En la lista Buckets, elija el bucket para el que desea habilitar una clave de bucket de S3.
3. En la lista Objects (Objetos), elija el nombre del objeto.
4. En la pestaña Details (Detalles), en Server-side encryption settings (Configuración de cifrado del lado del servidor), elija Edit (Editar).

   En Clave de bucket, verá el ajuste de la clave de bucket de S3 para su objeto. No puede editar esta configuración.

Uso de la AWS CLI

Para ver el ajuste de la clave de bucket de S3 de nivel de bucket,

Para utilizar este ejemplo, sustituya *`user input placeholder`* por su propia información.

```
aws s3api get-bucket-encryption --bucket amzn-s3-demo-bucket1
```

Para obtener más información, consulte get-bucket-encryption en la Referencia de comandos de la AWS CLI.

Para devolver la configuración de clave de bucket de S3 de nivel de objeto

Para utilizar este ejemplo, sustituya *user input placeholder* por su propia información.

```
aws s3api head-object --bucket amzn-s3-demo-bucket1 --key my_images.tar.bz2
```

Para obtener más información, consulte head-object en la Referencia de comandos de la AWS CLI.

Uso de la API de REST

Para ver el ajuste de la clave de bucket de S3 de nivel de bucket,

Para ver lar información de cifrado de un bucket, incluida la configuración de una clave de bucket de S3, utilice la operación `GetBucketEncryption`. El ajuste de la clave de bucket de S3 se ve en el cuerpo de la respuesta del elemento `ServerSideEncryptionConfiguration` con la configuración `BucketKeyEnabled`. Para obtener más información, consulte GetBucketEncryption en la Referencia de la API de Amazon S3.

Para ver la configuración de objeto de una clave de bucket de S3,

Para ver el estado de la clave de bucket de S3 de un objeto, utilice la operación `HeadObject`. `HeadObject` devuelve el encabezado de respuesta `x-amz-server-side-encryption-bucket-key-enabled` para mostrar si una clave de bucket de S3 está habilitada o deshabilitada para el objeto. Para obtener más información, consulte HeadObject en la Referencia de la API de Amazon S3.

Las siguientes operaciones de API también devuelven el encabezado de respuesta `x-amz-server-side-encryption-bucket-key-enabled` si se configura una clave de bucket de S3 para un objeto:

- PutObject
- PostObject
- CopyObject
- CreateMultipartUpload
- UploadPartCopy
- UploadPart
- CompleteMultipartUpload
- GetObject

Uso del cifrado del servidor de doble capa con claves de AWS KMS (DSSE-KMS)

Al utilizar el cifrado del servidor de doble capa con claves de AWS Key Management Service (AWS KMS) (DSSE-KMS), se aplican dos capas de cifrado a los objetos cuando se cargan en Amazon S3. El DSSE-KMS le ayuda a cumplir con mayor facilidad los estándares de conformidad que requieren que aplique un cifrado de varias capas a sus datos y tenga el control total de sus claves de cifrado.

Al utilizar el DSSE-KMS con un bucket de Amazon S3, las claves de AWS KMS deben estar en la misma región que el bucket. Además, cuando se solicita el DSSE-KMS para el objeto, la suma de comprobación de S3 que forma parte de los metadatos del objeto se almacena cifrada. Para obtener más información acerca de las sumas de comprobación, consulte Comprobación de la integridad de objetos.

La utilización de DSSE-KMS y AWS KMS keys conlleva cargos adicionales. Para obtener más información sobre los precios de DSSE-KMS, consulte Conceptos de AWS KMS key en la Guía para desarrolladores de AWS Key Management Service y Precios de AWS KMS.

> **Note**
>
> Las claves de bucket de S3 no son compatibles con DSSE-KMS.

Exigir el cifrado del servidor de doble capa con AWS KMS keys (DSSE-KMS)

Para exigir el cifrado del servidor de doble capa para todos los objetos en un bucket de Amazon S3 determinado, puede usar una política de bucket. Por ejemplo, la siguiente política de bucket deniega el permiso de carga de objeto (`s3:PutObject`) para todos, si la solicitud no incluye el encabezado `x-amz-server-side-encryption`, que solicita el cifrado del servidor con DSSE-KMS.

```
{
            "Version":"2012-10-17",
            "Id":"PutObjectPolicy",
            "Statement":[{
                  "Sid":"DenyUnEncryptedObjectUploads",
                  "Effect":"Deny",
                  "Principal":"*",
                  "Action":"s3:PutObject",
                  "Resource":"arn:aws:s3:::amzn-s3-demo-bucket1/*",
                  "Condition":{
                     "StringNotEquals":{
                        "s3:x-amz-server-side-encryption":"aws:kms:dsse"
```

```
                    }
                }
            }
        ]
    }
```

Temas

- Especificación del cifrado del servidor de doble capa con claves de AWS KMS (DSSE-KMS)

Especificación del cifrado del servidor de doble capa con claves de AWS KMS (DSSE-KMS)

> **Important**
>
> Amazon S3 aplica ahora el cifrado del servidor con claves administradas por Amazon S3 (SSE-S3) como el nivel básico de cifrado para cada bucket de Amazon S3. Desde el 5 de enero de 2023, todas las cargas de objetos nuevos a Amazon S3 se cifran automáticamente sin costo adicional y sin afectar al rendimiento. El estado de cifrado automático para la configuración de cifrado predeterminada en el bucket de S3 y para cargas de objetos nuevos está disponible en registros de AWS CloudTrail, Inventario de S3, Lente de almacenamiento de S3, la consola de Amazon S3 y como encabezado de respuesta a la API de Amazon S3 adicional en AWS Command Line Interface y los SDK de AWS. Para obtener más información, consulte Preguntas frecuentes del cifrado predeterminado.

Todos los buckets de Amazon S3 tienen el cifrado configurado de forma predeterminada y todos los objetos nuevos cargados en un bucket de S3 se cifran automáticamente en reposo. El cifrado del lado del servidor con claves administradas de Amazon S3 (SSE-S3) es la configuración de cifrado predeterminada para cada bucket de Amazon S3. Para usar otro tipo de cifrado, puede especificar el tipo de cifrado del servidor que se utilizará en las solicitudes PUT de S3 o puede establecer la configuración de cifrado predeterminada en el bucket de destino.

Si desea especificar un tipo de cifrado diferente en sus solicitudes PUT, puede utilizar el cifrado del servidor con claves de AWS Key Management Service (AWS KMS) (SSE-KMS), el cifrado del servidor de doble capa con claves de AWS KMS (DSSE-KMS) o el cifrado del servidor con claves proporcionadas por el cliente (SSE-C). Si desea establecer una configuración de cifrado predeterminada diferente en el bucket de destino, puede usar SSE-KMS o DSSE-KMS.

Puede aplicar cifrado cuando cargue un objeto nuevo o copie un objeto existente.

Puede especificar DSSE-KMS con la consola de Amazon S3, la API de REST de Amazon S3 y la AWS Command Line Interface (AWS CLI). Para obtener más información, consulte los siguientes temas.

> **Note**
>
> Puede utilizar AWS KMS keys de varias regiones en Amazon S3. No obstante, Amazon S3 trata las claves de varias regiones como si fueran claves de una sola región y no utiliza las características de varias regiones de la clave. Para obtener más información, consulte Uso de claves de varias regiones en la Guía para desarrolladores de AWS Key Management Service.

> **Note**
>
> Si desea utilizar una clave de KMS propiedad de una cuenta diferente, primero debe tener permiso para utilizar la clave. Para obtener más información sobre los permisos entre cuentas para las claves de KMS, consulte Crear claves de KMS que otras cuentas puedan utilizar en la Guía para desarrolladores de AWS Key Management Service.

Uso de la consola de S3

En esta sección se describe cómo configurar o cambiar el tipo de cifrado de un objeto para utilizar el cifrado del servidor de doble capa con claves de AWS Key Management Service (AWS KMS) (DSSE-KMS) mediante la consola de Amazon S3.

> **Note**
>
> - Si cambia el método de cifrado de un objeto, se crea un nuevo objeto para reemplazar al antiguo. Si el control de versiones de S3 está activado, se crea una nueva versión del objeto y el objeto existente se convierte en una versión anterior. El rol que cambia la propiedad también se convierte en el propietario del nuevo objeto o (versión del objeto).
> - Si cambia el tipo de cifrado de un objeto que tiene etiquetas definidas por el usuario, debe tener el permiso `s3:GetObjectTagging`. Si va a cambiar el tipo de cifrado de un objeto que no tiene etiquetas definidas por el usuario pero que tiene un tamaño superior a 16 MB, también debe tener el permiso `s3:GetObjectTagging`.

Si la política de bucket de destino deniega la acción `s3:GetObjectTagging`, el tipo de cifrado del objeto se actualizará, pero las etiquetas definidas por el usuario se eliminarán del objeto y aparecerá un error.

Para añadir o cambiar el cifrado de un objeto

1. Inicie sesión en la AWS Management Console y abra la consola de Amazon S3 en https://console.aws.amazon.com/s3/.
2. En el panel de navegación izquierdo, elija Instancias.
3. En la lista Buckets, seleccione el nombre del bucket que contiene los objetos que desea cifrar.
4. En la lista Objetos, seleccione la casilla de verificación junto al objeto que desea agregar cifrado o cuyo cifrado desea modificar.

   Aparece la página de detalles del objeto, con varias secciones que muestran las propiedades del objeto.
5. Elija la pestaña Propiedades.
6. Desplácese hacia abajo hasta la sección Cifrado predeterminado y elija Editar.

   Se abre la página Editar cifrado del lado del servidor.
7. En Tipo de cifrado, seleccione Cifrado del servidor de doble capa con claves de AWS Key Management Service (DSSE-KMS).
8. En Clave de AWS KMS, siga una de las siguientes opciones para elegir su clave de KMS:

   - Para seleccionar en una lista de claves de KMS disponibles, marque Elija entre sus claves de AWS KMS keys y seleccione su clave de KMS en la lista de claves disponibles.

     En esta lista aparecen tanto la Clave administrada de AWS (`aws/s3`) como las claves administradas por el cliente. Para obtener más información acerca de las claves administradas por el cliente, consulte Claves de cliente y claves de AWS en la Guía para desarrolladores de AWS Key Management Service.
   - Para introducir el ARN de la clave de KMS, elija Introducir el ARN de la clave de AWS KMS key e introduzca el ARN de la clave de KMS en el campo que aparece.
   - Para crear una nueva clave administrada por el cliente en la consola de AWS KMS, elija Crear una clave de KMS.

Para obtener más información acerca de cómo crear una AWS KMS key, consulte Creación de claves en la AWS Key Management Service Guía para desarrolladores.

> **Important**
>
> Solo puede utilizar las claves de KMS que estén disponibles en la misma Región de AWS del bucket. La consola de Amazon S3 solo muestra las primeras 100 claves de KMS de la misma región del bucket. Para utilizar una clave de KMS que no aparezca en la lista, debe introducir el ARN de la clave de KMS. Si desea utilizar una clave de KMS propiedad de una cuenta de diferente, primero debe tener permiso para utilizar la clave y, después, debe introducir el ARN de la clave de KMS.
>
> Amazon S3 admite solo claves KMS de cifrado simétricas y no claves KMS asimétricas. Para obtener más información, consulte Identificación de claves de KMS asimétricas en la Guía para desarrolladores de AWS Key Management Service.

9. En Clave de bucket, seleccione Desactivar. Las claves de bucket de S3 no son compatibles con DSSE-KMS.
10. Elija Guardar cambios.

> **Note**
>
> En esta acción se aplica el cifrado a todos los objetos especificados. Al cifrar carpetas, espere a que finalice la operación de guardado para agregar nuevos objetos a la carpeta.

Uso de la API de REST

Cuando cree un objeto, es decir, cuando cargue un objeto nuevo o copie uno existente, podrá especificar el uso del cifrado del servidor de doble capa con AWS KMS keys (DSSE-KMS) para cifrar los datos. Para ello, añada el encabezado `x-amz-server-side-encryption` a la solicitud. Configure el valor del encabezado para el algoritmo de cifrado `aws:kms:dsse`. Amazon S3 confirma que su objeto se ha almacenado con cifrado DSSE-KMS al devolver el encabezado de respuesta `x-amz-server-side-encryption`.

Si especifica el encabezado `x-amz-server-side-encryption` con un valor de `aws:kms:dsse`, también puede utilizar los siguientes encabezados de solicitud:

- `x-amz-server-side-encryption-aws-kms-key-id: SSEKMSKeyId`
- `x-amz-server-side-encryption-context: SSEKMSEncryptionContext`

Temas

- Operaciones de la API de REST de Amazon S3 que admiten DSSE-KMS
- Contexto de cifrado (x-amz-server-side-encryption-context)
- ID de clave de AWS KMS (x-amz-server-side-encryption-aws-kms-key-id)

Operaciones de la API de REST de Amazon S3 que admiten DSSE-KMS

Las siguientes operaciones de la API de REST aceptan los encabezados de solicitud `x-amz-server-side-encryption`, `x-amz-server-side-encryption-aws-kms-key-id` y `x-amz-server-side-encryption-context`.

- PutObject: cuando cargue datos mediante la operación de la API PUT, puede especificar estos encabezados de solicitud.
- CopyObject: al copiar un objeto, tiene un objeto de origen y otro de destino. Al pasar encabezados DSSE-KMS con la operación `CopyObject`, estos se aplican solo al objeto de destino. Cuando copie un objeto existente, independientemente de si el objeto de origen está cifrado o no, el objeto de destino no estará cifrado, a no ser que solicite explícitamente el cifrado del lado del servidor.
- Objeto POST: cuando utilice una operación `POST` para cargar un objeto, en vez de proporcionar los encabezados de solicitud, debe proporcionar la misma información en los campos del formulario.
- CreateMultipartUpload: cuando cargue objetos grandes mediante la carga multiparte, puede especificar estos encabezados en la solicitud `CreateMultipartUpload`.

Los encabezados de respuesta de las siguientes operaciones de API de REST devuelven el encabezado `x-amz-server-side-encryption` cuando un objeto se almacena con el cifrado del servidor.

- PutObject
- CopyObject
- Objeto POST
- CreateMultipartUpload

- UploadPart
- UploadPartCopy
- CompleteMultipartUpload
- GetObject
- HeadObject

> **Important**
>
> - Todas las solicitudes GET y PUT para un objeto protegido por AWS KMS generarán un error si no las realiza con la capa de sockets seguros (SSL), la seguridad de la capa de transporte (TLS) o Signature Version 4.
> - Si su objeto utiliza DSSE-KMS, no envíe encabezados de solicitud de cifrado para solicitudes GET y HEAD porque, de lo contrario, obtendrá un error HTTP 400 (Bad Request).

Contexto de cifrado (**`x-amz-server-side-encryption-context`**)

Si especifica `x-amz-server-side-encryption:aws:kms:dsse`, la API de Amazon S3 admitirá un contexto de cifrado con el encabezado `x-amz-server-side-encryption-context`.
Un contexto de cifrado es un conjunto definido de pares clave-valor que contienen información contextual adicional sobre los datos.

Amazon S3 utiliza automáticamente el nombre de recurso de Amazon (ARN) del objeto como par del contexto de cifrado; por ejemplo, `arn:aws:s3:::`*`object_ARN`*.

Si lo desea, puede proporcionar un par de contexto de cifrado adicional mediante el encabezado `x-amz-server-side-encryption-context`. No obstante, dado que el contexto de cifrado no está cifrado, asegúrese de no incluir información confidencial. Amazon S3 almacena este par de claves adicional junto con el contexto de cifrado predeterminado.

Para obtener información sobre el contexto de cifrado en Amazon S3, consulte Contexto de cifrado. Para obtener información general sobre el contexto de cifrado, consulte Conceptos de AWS Key Management Service: contexto de cifrado en la Guía para desarrolladores de AWS Key Management Service.

ID de clave de AWS KMS (**x-amz-server-side-encryption-aws-kms-key-id**)

Puede utilizar el encabezado `x-amz-server-side-encryption-aws-kms-key-id` para especificar el ID de la clave administrada por el cliente utilizado para proteger los datos. Si especifica el encabezado `x-amz-server-side-encryption:aws:kms:dsse`, pero no proporciona el encabezado `x-amz-server-side-encryption-aws-kms-key-id`, Amazon S3 utiliza la Clave administrada de AWS (`aws/s3`) para proteger los datos. Si desea utilizar una clave administrada por el cliente, debe proporcionar el encabezado `x-amz-server-side-encryption-aws-kms-key-id` de dicha clave.

> **Important**
>
> Cuando utilice una AWS KMS key para el cifrado en el lado del servidor en Amazon S3, debe elegir una clave de cifrado de KMS simétrica. Amazon S3 solo admite claves KMS de cifrado simétricas. Para obtener más información sobre estas claves, consulte Symmetric encryption KMS keys (Claves de KMS de cifrado simétricas) en la Guía para desarrolladores de AWS Key Management Service.

Uso de la AWS CLI

Cuando se carga un objeto nuevo o se copia uno existente, puede especificar que se use DSSE-KMS para cifrar los datos. Para ello, añada el parámetro `--server-side-encryption aws:kms:dsse` a la solicitud. Utilice el parámetro `--ssekms-key-id` *`example-key-id`* para agregar la clave de AWS KMS administrada de cliente que ha creado. Si especifica `--server-side-encryption aws:kms:dsse`, pero no proporciona un ID de clave de AWS KMS, Amazon S3 utilizará la clave administrada de AWS (`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
```

Puede cifrar un objeto no cifrado para usar DSSE-KMS volviendo a copiar el objeto en su lugar.

```
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
```

Uso de cifrado en el lado del servidor con claves proporcionadas por el cliente (SSE-C)

El cifrado en el servidor consiste en proteger los datos en reposo. El cifrado en el servidor solo cifra los datos de objetos, no los metadatos de objetos. Mediante el cifrado del servidor con claves proporcionadas por el cliente (SSE-C) puede almacenar los datos cifrados con sus propias claves de cifrado. Con la clave de cifrado que proporcione como parte de su solicitud, Amazon S3 administra tanto el cifrado de datos, al escribir en los discos, como el descifrado de datos, al obtener acceso a los objetos. Por tanto, no ha de mantener ningún código para llevar a cabo el cifrado y el descifrado de los datos. Lo único que tiene que hacer es administrar las claves de cifrado que proporcione.

Cuando carga un objeto, Amazon S3 usa la clave de cifrado facilitada para aplicar un cifrado AES-256 a los datos. A continuación, Amazon S3 elimina la clave de cifrado de la memoria. Al recuperar un objeto, debe facilitar la misma clave de cifrado como parte de la solicitud. En primer lugar, Amazon S3 comprueba que la clave de cifrado proporcionada coincida, y a continuación descifra el objeto antes de devolverle los datos del mismo.

El uso de SSE-C no tiene costes adicionales. Sin embargo, las solicitudes de configuración y uso de SSE-C incurren en cargos estándar de solicitud de Amazon S3. Para obtener información acerca de los precios, consulte Precios de Amazon S3.

> **Note**
>
> Amazon S3 no almacena la clave de cifrado que facilite. En su lugar, almacena un valor de código de autenticación de mensajes basado en hash (HMAC) discontinuo aleatorio de la clave de cifrado para validar las solicitudes futuras. El valor HMAC "salted" no se puede usar para derivar el valor de la clave de cifrado ni para descifrar los contenidos del objeto cifrado. Esto implica que, si pierde la clave de cifrado, habrá perdido el objeto.

La replicación de S3 admite objetos cifrados con SSE-C. Para obtener más información sobre la replicación de objetos cifrados, consulte the section called “Replicar objetos cifrados”.

Para obtener más información acerca de SSE-C, vea los temas siguientes.

Temas

- Información general de SSE-C
- Exigir y restringir SSE-C

- URL prefirmadas y SSE-C
- Especificación del cifrado del lado del servidor con claves proporcionadas por el cliente (SSE-C)

Información general de SSE-C

En esta sección, se proporciona información general acerca del SSE-C. Cuando utilice SSE-C, tenga en cuenta las siguientes consideraciones.

- Debe utilizar HTTPS.

> **Important**
>
> Amazon S3 rechaza cualquier solicitud que se realice por HTTP cuando use SSE-C. Por motivos de seguridad, le recomendamos que tenga en cuenta que cualquier clave que envíe por error sobre HTTP podría estar en peligro. Descarte la clave y practique la rotación apropiada.

- La etiqueta de entidad (ETag) de la respuesta no es el MD5 de los datos del objeto.
- Debe administrar el mapeo de qué clave de cifrado se utiliza para cifrar cada objeto. Amazon S3 no almacena claves de cifrado. Usted debe responsabilizarse de realizar un seguimiento de qué clave de cifrado proporciona para cada objeto.
  - Si su bucket tiene activado el control de versiones, cada versión del objeto que cargue utilizando esta característica tendrá su propia clave de cifrado. Usted debe responsabilizarse de realizar un seguimiento de qué clave de cifrado se ha utilizado en cada versión del objeto.
  - Dado que es usted quien administra las claves de cifrado en el cliente, ha de administrar todas las garantías adicionales, como la rotación de claves, en el lado del cliente.

> **Warning**
>
> Si pierde la clave de cifrado, todas las solicitudes `GET` de un objeto sin su clave de cifrado provoca un error y pierde el objeto.

Exigir y restringir SSE-C

Para requerir SSE-C para todos los objetos en un bucket de Amazon S3 particular, utilice una política de bucket.

Por ejemplo, la siguiente política de bucket deniega la carga de objetos (`s3:PutObject`) para todas las solicitudes que no incluyan el encabezado `x-amz-server-side-encryption-customer-algorithm` que solicita SSE-C.

```
{
    "Version": "2012-10-17",
    "Id": "PutObjectPolicy",
    "Statement": [
        {
            "Sid": "RequireSSECObjectUploads",
            "Effect": "Deny",
            "Principal": "*",
            "Action": "s3:PutObject",
            "Resource": "arn:aws:s3:::amzn-s3-demo-bucket/*",
            "Condition": {
                "Null": {
                    "s3:x-amz-server-side-encryption-customer-algorithm": "true"
                }
            }
        }
    ]
}
```

También puede utilizar una política para restringir el cifrado del lado del servidor para todos los objetos en un bucket de Amazon S3 particular. Por ejemplo, la siguiente política de bucket deniega el permiso de carga de objeto (`s3:PutObject`) para todos, si la solicitud incluye el encabezado `x-amz-server-side-encryption-customer-algorithm` que solicita SSE-C.

```
{
    "Version": "2012-10-17",
    "Id": "PutObjectPolicy",
    "Statement": [
        {
            "Sid": "RestrictSSECObjectUploads",
            "Effect": "Deny",
            "Principal": "*",
            "Action": "s3:PutObject",
            "Resource": "arn:aws:s3:::amzn-s3-demo-bucket/*",
            "Condition": {
                "Null": {
                    "s3:x-amz-server-side-encryption-customer-algorithm": "false"
```

```
                }
            }
        }
    ]
}
```

> **Important**
>
> Si utiliza una política de bucket para exigir SSE-C en `s3:PutObject`, debe incluir el encabezado `x-amz-server-side-encryption-customer-algorithm` en todas las solicitudes de carga multiparte (CreateMultipartUpload, UploadPart y CompleteMultipartUpload).

URL prefirmadas y SSE-C

Puede generar una URL prefirmada, que podrá utilizar para operaciones como la carga de un nuevo objeto, la recuperación de un objeto existente o la recuperación de metadatos de objetos. Las URL prefirmadas se usan con el SSE-C de la siguiente forma:

- Al crear una URL prefirmada, debe especificar el algoritmo utilizando el encabezado `x-amz-server-side-encryption-customer-algorithm` en el cálculo de la firma.
- Al usar la URL prefirmada para cargar un objeto nuevo, recuperar un objeto existente o recuperar solo metadatos de objetos, debe facilitar todos los encabezados de cifrado en su solicitud de aplicación cliente.

> **Note**
>
> Para objetos no SSE-C, puede generar una URL prefirmada y pegar dicha URL directamente en un navegador para acceder a los datos.
> No obstante, no puede hacer esto en objetos SSE-C porque además de la URL prefirmada también debe incluir encabezamientos de HTTP específicos de objetos SSE-C Por tanto, puede usar las URL prefirmadas para objetos SSE-C solo mediante programación.

Para obtener más información acerca de URL prefirmadas, consulte the section called “Uso de URL prefirmadas”.

Especificación del cifrado del lado del servidor con claves proporcionadas por el cliente (SSE-C)

En el momento de la creación de objetos con la API de REST, puede especificar el cifrado del lado del servidor con claves proporcionadas por el cliente (SSE-C). Cuando utilice SSE-C, deberá proporcionar información sobre la clave de cifrado mediante los siguientes encabezados de solicitud.

Nombre	Descripción
`x-amz-server-side-encryption-customer-algorithm`	Use este encabezado para especificar el algoritmo de cifrado. El valor del encabezado ha de ser `AES256`.
`x-amz-server-side-encryption-customer-key`	Use este encabezado para facilitar la clave de cifrado de 256 bits con codificación base64 para que Amazon S3 pueda usarla para cifrar o descifrar los datos.
`x-amz-server-side-encryption-customer-key-MD5`	Use este encabezado para facilitar el resumen MD5 de 128 bits con codificación en base64 de la clave de cifrado, según la RFC 1321. Amazon S3 usa este encabezado para comprobar la integridad del mensaje y garantizar que la clave de cifrado se haya transmitido sin errores.

Puede usar las bibliotecas de encapsulamiento del AWS SDK para agregar estos encabezados a su solicitud. Si lo necesita, también puede realizar las llamadas a la API de REST de Amazon S3 directamente en su aplicación.

> **Note**
>
> No puede usar la consola de Amazon S3 para cargar un objeto y solicitar SSE-C. Tampoco puede usar la consola para actualizar (por ejemplo: cambiar la clase de almacenamiento o agregar metadatos) un objeto existente almacenado con SSE-C.

Uso de la API de REST

API de REST de Amazon S3 que admiten SSE-C

Las siguientes API de Amazon S3 admiten el cifrado del lado del servidor con claves de cifrado (SSE-C) proporcionadas por el cliente.

- Operación GET: cuando recupera datos con la API GET (consulte GET Object), puede especificar los encabezados de solicitud.
- Operación HEAD: para recuperar metadatos de objetos con la API HEAD (consulte HEAD Object), puede especificar estos encabezados de solicitud.
- Operación PUT: cuando carga datos con la API de PUT Object (consulte PUT Object), puede especificar estos encabezados de solicitud.
- Carga multiparte: al cargar objetos grandes mediante la API de carga multiparte, puede especificar estos encabezados. Debe especificar estos encabezados en la solicitud inicial (consulte Iniciar carga multiparte) y en cada solicitud de carga de partes subsiguiente (consulte Cargar parte o Cargar parte - Copia). Para cada solicitud de carga de parte, la información de cifrado ha de ser la misma que la facilitada en la solicitud inicial de la carga multiparte.
- Operación POST: cuando utiliza una operación POST para cargar un objeto (consulte POST Object), en vez de proporcionar los encabezados de solicitud, debe proporcionar la misma información en los campos del formulario.
- Operación Copy: cuando copia un objeto (consulte PUT Object - Copy), tiene un objeto de origen y uno de destino.
  - Si quiere que el objeto de destino se cifre mediante el cifrado del lado del servidor con claves administradas por AWS, debe proporcionar el encabezado de solicitud `x-amz-server-side-encryption`.
  - Si quiere que el objeto objetivo se cifre mediante SSE-C, debe facilitar información de cifrado mediante los tres encabezados descritos en la tabla anterior.
  - Si el objeto de origen está cifrado con SSE-C, debe facilitar la información de la clave de cifrado mediante los siguientes encabezados, de modo que Amazon S3 puede descifrar el objeto para copiarlo.

Nombre	Descripción
`x-amz-copy-source-server-side-encryption-customer-algorithm`	Incluya este encabezado para especificar el algoritmo que debe usar Amazon S3 para descifrar el objeto de origen. Este valor debe ser `AES256`.
`x-amz-copy-source-server-side-encryption-customer-key`	Incluya este encabezado para facilitar la clave de cifrado con codificación base64 para que Amazon S3 la utilice para descifrar el objeto de origen. Esta clave de cifrado debe ser la que proporcionó a Amazon S3 al crear el objeto de origen. De lo contrario, Amazon S3 no puede descifrar el objeto.
`x-amz-copy-source-server-side-encryption-customer-key-MD5`	Incluya este encabezado para facilitar el resumen MD5 de 128 bits con codificación en base64 de la clave de cifrado, según la RFC 1321.

Uso de los AWS SDK para especificar SSE-C en las operaciones PUT, GET, Head y Copy

En el siguiente ejemplo se muestra cómo solicitar el cifrado del lado del servidor con claves proporcionadas por el cliente (SSE-C) para objetos. En los ejemplos se realizan las siguientes operaciones. Cada operación muestra cómo especificar encabezados relacionados con el SSE-C en la solicitud:

- Put object: carga un objeto y solicita el cifrado del lado del servidor mediante una clave de cifrado proporcionada por un cliente.
- Get object: descarga el objeto que se cargó en el paso anterior. En la solicitud, proporciona la misma información de cifrado que proporcionó al cargar el objeto. Amazon S3 necesita esta información para descifrar el objeto de modo que pueda devolvérselo.
- Get object metadata: recupera los metadatos del objeto. Proporciona la misma información de cifrado usada al crear el objeto.

- Copy object: realiza una copia del objeto cargado previamente. Dado que el objeto de origen se almacena mediante SSE-C, usted debe proporcionar la información de cifrado en su solicitud de copia. De forma predeterminada, Amazon S3 cifra la copia del objeto solo si lo solicita explícitamente. En este ejemplo se indica a Amazon S3 que almacene una copia cifrada del objeto.

Java

> **Note**
>
> Este ejemplo muestra cómo cargar un objeto en una operación única. Cuando utiliza la API de carga multiparte para cargar objetos grandes, brinda información de cifrado como se muestra en el siguiente ejemplo. Para ver ejemplos de cargas multiparte que utilizan AWS SDK for Java, consulte Carga de un objeto con la carga multiparte.

Para añadir la información de cifrado necesaria, incluya una `SSECustomerKey` en su solicitud. Para obtener más información sobre la clase `SSECustomerKey` , consulte la sección REST API (API de REST).

Para obtener más información acerca de SSE-C, consulte Uso de cifrado en el lado del servidor con claves proporcionadas por el cliente (SSE-C). Para obtener instrucciones sobre cómo crear y probar una muestra funcional, consulte Introducción en la Guía del desarrollador de AWS SDK for Java.

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**
>
> Para ver ejemplos de cómo cargar objetos grandes con la API de carga multiparte, consulte Carga de un objeto con la carga multiparte y Uso de los AWS SDK (API de bajo nivel).

Para obtener más información acerca de SSE-C, consulte Uso de cifrado en el lado del servidor con claves proporcionadas por el cliente (SSE-C). Para obtener información acerca de cómo configurar y ejecutar ejemplos de código, consulte Introducción al SDK de AWS para .NET en la Guía para desarrolladores del SDK de AWS para .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);
        }
    }
}
```

Uso de los AWS SDK para especificar SSE-C en las cargas multiparte

En el ejemplo de la sección anterior se muestra cómo solicitar cifrado del lado del servidor con claves proporcionadas por el cliente (SSE-C) en operaciones PUT, GET, Head y Copy. En esta sección se describen otras API de Amazon S3 que admiten SSE-C.

Java

Para cargar objetos grandes, puede utilizar la API de carga multiparte (consulte Carga y copia de objetos con la carga multiparte). Puede usar API de nivel alto o de nivel bajo para cargar objetos grandes. Estas API admiten los encabezados relacionados con el cifrado en la solicitud.

- Cuando utiliza la API `TransferManager` de alto nivel, usted proporciona los encabezados específicos del cifrado en la `PutObjectRequest` (consulte Carga de un objeto con la carga multiparte).
- Al usar la API de bajo nivel, proporcionará información relacionada con el cifrado en la `InitiateMultipartUploadRequest`, seguida por información de cifrado idéntica en cada

`UploadPartRequest`. No necesita proporcionar encabezados específicos de cifrado en su `CompleteMultipartUploadRequest`. Para ver ejemplos, consulte Uso de los AWS SDK (API de bajo nivel).

En el siguiente ejemplo se usa `TransferManager` para crear objetos y se muestra cómo facilitar la información relacionada con SSE-C. En el ejemplo se realiza lo siguiente:

- Crea un objeto mediante el método `TransferManager.upload()`. En la instancia `PutObjectRequest`, proporciona la información de la clave de cifrado para solicitar que Amazon S3 cifre el objeto utilizando la clave de facilitada por el cliente.
- Realiza una copia del objeto llamando al método `TransferManager.copy()`. El ejemplo indica a Amazon S3 que cifre la copia del objeto con una nueva `SSECustomerKey`. Dado que el objeto de origen está cifrado con SSE-C, la `CopyObjectRequest` también facilita la clave de cifrado del objeto de origen, de modo que Amazon S3 puede descifrar el objeto antes de copiarlo.

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

Para cargar objetos grandes, puede utilizar la API de carga multiparte (consulte Carga y copia de objetos con la carga multiparte).AWS SDK para .NET proporciona API de nivel alto o bajo para cargar objetos de gran tamaño. Estas API admiten los encabezados relacionados con el cifrado en la solicitud.

- Cuando utiliza la API de `Transfer-Utility` de alto nivel, usted proporciona los encabezados específicos del cifrado en `TransferUtilityUploadRequest` como se muestra. Para ver ejemplos de código, consulte Carga de un objeto con la carga multiparte.

```
TransferUtilityUploadRequest request = new TransferUtilityUploadRequest()
{
    FilePath = filePath,
    BucketName = existingBucketName,
    Key = keyName,
    // Provide encryption information.
    ServerSideEncryptionCustomerMethod =
 ServerSideEncryptionCustomerMethod.AES256,
    ServerSideEncryptionCustomerProvidedKey = base64Key,
};
```

- Al usar la API de bajo nivel, proporcionará información relacionada con el cifrado en la solicitud de inicio de la carga multiparte, seguida por información de cifrado idéntica en las solicitudes de carga de partes subsiguientes. No necesita proporcionar encabezados específicos de cifrado en su solicitud de carga multiparte completa. Para ver ejemplos, consulte Uso de los AWS SDK (API de bajo nivel).

  A continuación se muestra un ejemplo de carga multiparte de bajo nivel que hace una copia de un objeto grande existente. En el ejemplo, el objeto que se copiará se guarda en Amazon S3 mediante el SSE-C y usted también desea usar el SSE-C para guardar el objeto de destino. En el ejemplo, hará lo siguiente:

  - Inicie una solicitud de carga multiparte proporcionando una clave de cifrado y la información relacionada.
  - Proporcione las claves de cifrado del objeto de origen y de destino, y la información relacionada en `CopyPartRequest`.
  - Recupere los metadatos del objeto para obtener el tamaño del objeto de origen que se copiará.
  - Cargue los objetos en partes de 5 MB.

  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);
            }
        }
    }
}
```

Protección de los datos con el cifrado del cliente

El cifrado del cliente es el acto de cifrar los datos de manera local para ayudar a garantizar su seguridad en tránsito y en reposo. Para cifrar los objetos antes de enviarlos a Amazon S3, utilice el cliente de cifrado de Amazon S3. Si cifra sus objetos de esta manera, no estarán expuestos a terceros, ni siquiera AWS. Amazon S3 recibe los objetos ya cifrados; Amazon S3 no interviene en el cifrado ni el descifrado de los objetos. Puede utilizar tanto el cliente de cifrado de Amazon S3 como el cifrado del lado del servidor para cifrar sus datos. Cuando envía objetos cifrados a Amazon S3, Amazon S3 no reconoce los objetos como cifrados, solo detecta los objetos típicos.

El cliente de cifrado de Amazon S3 funciona como intermediario entre usted y Amazon S3. Tras crear una instancia del cliente de cifrado de Amazon S3, sus objetos se cifran y descifran automáticamente como parte de sus solicitudes `PutObject` y `GetObject` de Amazon S3. Todos sus objetos se cifran con una clave de datos única. El cliente de cifrado de Amazon S3 no usa claves de bucket ni interactúa con ellas, incluso si especifica una clave de KMS como clave de empaquetado.

La Guía para desarrolladores del cliente de cifrado de Amazon S3 se centra en las versiones 3.0 y posteriores del cliente de cifrado de Amazon S3. Para obtener más información, consulte ¿Qué es el cifrado del cliente de Amazon S3? en la Guía para desarrolladores de cifrado del cliente de Amazon S3.

Para obtener más información acerca de las versiones anteriores del cliente de cifrado de Amazon S3, consulte la Guía para desarrolladores del SDK de AWS de su lenguaje de programación.

- AWS SDK for Java
- AWS SDK for .NET
- AWS SDK for Go
- AWS SDK for PHP
- AWS SDK for Ruby
- AWS SDK for C++

Privacidad del tráfico entre redes

Este tema describe cómo Amazon S3 protege las conexiones desde el servicio a otras ubicaciones.

Tráfico entre el servicio y las aplicaciones y clientes locales

Las siguientes conexiones se pueden combinar con AWS PrivateLink para proporcionar conectividad entre la red privada y AWS:

- Una conexión de Site-to-Site VPN de AWS. Para obtener más información, consulte ¿Qué es AWS Site-to-Site VPN?
- Una conexión de AWS Direct Connect. Para obtener más información, consulte ¿Qué es AWS Direct Connect?

El acceso a Amazon S3 a través de la red se realiza mediante las API publicadas por AWS. Los clientes deben admitir el protocolo de seguridad de la capa de transporte (TLS) 1.2. Nosotros recomendamos TLS 1.3. Los clientes también deben admitir conjuntos de cifrado con confidencialidad directa total (PFS) tales como Ephemeral Diffie-Hellman (DHE) o Elliptic Curve Diffie-Hellman Ephemeral (ECDHE). La mayoría de los sistemas modernos como Java 7 y posteriores son compatibles con estos modos. Además, debe firmar las solicitudes con un ID de clave de acceso y una clave de acceso secreta que estén asociados a una entidad principal de IAM, o bien, puede usar AWS Security Token Service (STS) para generar credenciales de seguridad temporales a la hora de firmar solicitudes.

Tráfico entre recursos de AWS en la misma región

Un punto de conexión de la nube privada virtual (Virtual Private Cloud, VPC) de Amazon S3 es una entidad lógica dentro de una VPC que permite la conectividad solo a Amazon S3. La VPC direcciona las solicitudes a Amazon S3 y vuelve a direccionar las respuestas a la VPC. Para obtener más información, consulte Puntos de enlace de la VPC en la guía del usuario de VPC. Para consultar ejemplos de políticas de bucket que puede utilizar para controlar el acceso al bucket de S3 desde los puntos de enlace de la VPC, consulte Control del acceso desde puntos de enlace de la VPC con políticas de bucket.

AWS PrivateLink para Amazon S3

Con AWS PrivateLink para Amazon S3, puede aprovisionar puntos de enlace de la VPC de la interfaz (puntos de enlace de la interfaz) en su Virtual Private Cloud (VPC). A estos puntos de enlace se puede acceder directamente desde las aplicaciones que se encuentran en las instalaciones a través de la VPN y AWS Direct Connect, o bien, en una Región de AWS diferente mediante la interconexión de VPC.

Los puntos de enlace de la interfaz se representan mediante una o más interfaces de red elásticas (elastic network interfaces, ENI) a las que se asignan direcciones IP privadas desde subredes de la VPC. Las solicitudes a Amazon S3 sobre puntos de conexión de la interfaz permanecen en la red de Amazon. Asimismo, puede acceder a los puntos de conexión de la interfaz en su VPC desde aplicaciones en las instalaciones a través de AWS Direct Connect oAWS Virtual Private Network (AWS VPN). Para obtener más información sobre cómo conectar la VPC a la red en las instalaciones, consulte la AWS Direct ConnectGuía del usuario de y la AWS Site-to-Site VPNGuía del usuario de .

Para obtener más información sobre los puntos de enlace de la interfaz, consulte Puntos de enlace de la VPC de la interfaz (AWS PrivateLink) en la Guía de AWS PrivateLink.

Temas

Tipos de puntos de enlace de la VPC para Amazon S3

Puede utilizar dos tipos de puntos de conexión de VPC para acceder a Amazon S3: puntos de conexión de la puerta de enlace y puntos de conexión de la interfaz (mediante AWS PrivateLink). Un punto de conexión de gateway es una gateway que se especifica en la tabla de enrutamiento para acceder a Amazon S3 desde su VPC a través de la red de AWS. Los puntos de conexión de la interfaz extienden la funcionalidad de los puntos de conexión de la puerta de enlace a través de direcciones IP privadas para enviar solicitudes a Amazon S3 desde su VPC, el sistema en las instalaciones u otra VPC en otra Región de AWS mediante la interconexión de VPC o AWS Transit Gateway. Para obtener más información, consulte ¿Qué es una interconexión de VPC? y Transit Gateway frente a interconexión de VPC.

Los puntos de enlace de la interfaz son compatibles con los puntos de enlace de gateway. Si tiene un punto de conexión de gateway existente en la VPC, puede utilizar ambos tipos de puntos de enlace en la misma VPC.

Puntos de enlace de gateway para Amazon S3	Puntos de enlace de interfaz para Amazon S3
En ambos casos, el tráfico de red permanece en la red de AWS.	
Utilice direcciones IP públicas de Amazon S3	Utilice direcciones IP privadas de su VPC para acceder a Amazon S3
Utilizar los mismos nombres DNS de Simple Storage Service (Amazon S3)	Requerir nombres DNS de Simple Storage Service (Amazon S3) específicos de punto de conexión
No permite el acceso desde las instalaciones	Permitir el acceso desde las instalaciones
No permite el acceso desde otra Región de AWS	Permite el acceso desde una VPC en otra Región de AWS mediante el uso de emparejamiento de VPC o AWS Transit Gateway
No facturado	Facturado

Para obtener más información acerca de los puntos de enlace de gateway, consulte Puntos de enlace de la VPC de gateway en la Guía del usuario de AWS PrivateLink.

Restricciones y límites de AWS PrivateLink para Amazon S3

Los límites de VPC se aplican a AWS PrivateLink para Amazon S3. Para obtener más información, consulte Consideraciones de los puntos de conexión de la interfaz y Cuotas de AWS PrivateLink en la Guía de AWS PrivateLink. Además, se aplican las siguientes restricciones.

AWS PrivateLink para Amazon S3 no admite lo siguiente:

- Puntos de conexión del estándar federal de procesamiento de información (FIPS)
- Puntos de enlace de sitio web
- Puntos de enlace global heredado
- Puntos de conexión de S3 guion región

- Puntos de conexión de doble pila en Amazon S3
- Usar CopyObject o UploadPartCopy entre buckets en diferentes Regiones de AWS
- Seguridad de la capa de transporte (TLS) 1.1

Creación de un punto de conexión de VPC

Para crear un punto de conexión de interfaz de VPC, consulte Crear un punto de conexión de VPC en la Guía de AWS PrivateLink.

Acceso a los puntos de enlace de la interfaz de Amazon S3

Cuando se crea un punto de conexión de interfaz, Amazon S3 genera dos tipos de nombres DNS de S3 específicos del punto de conexión: regional y zonal.

- Un nombre de DNS regional incluye un ID único de punto de conexión de VPC, un identificador de servicio, la Región de AWS y `vpce.amazonaws.com` en su nombre. Por ejemplo, para el Id. de punto de conexión de la VPC *`vpce-1a2b3c4d`*, el nombre DNS generado podría ser similar a *`vpce-1a2b3c4d-5e6f`*`.s3.us-east-1.vpce.amazonaws.com`.
- Un nombre de DNS zonal incluye la zona de disponibilidad, por ejemplo, *`vpce-1a2b3c4d-5e6f`*`-`**`us-east-1a.`**`s3.us-east-1.vpce.amazonaws.com`. Puede utilizar esta opción si la arquitectura aísla Zonas de disponibilidad. Por ejemplo, podría usarlo para la contención de fallos o para reducir los costos de transferencia de datos regionales.

Los nombres DNS S3 específicos de los puntos de enlace se pueden resolver desde el dominio DNS público de S3.

DNS privado

Las opciones de DNS privado para los puntos de conexión de la interfaz de VPC simplifican el enrutamiento del tráfico de S3 a través de los puntos de conexión de VPC y le ayudan a aprovechar la ruta de red más económica disponible para su aplicación. Puede usar las opciones de DNS privadas para dirigir el tráfico de S3 regional sin actualizar sus clientes de S3 para que usen los nombres DNS específicos de los puntos de conexión de sus interfaces ni administrar la infraestructura de DNS. Con los nombres DNS privados habilitados, las consultas de DNS de S3 regional se resuelven en las direcciones IP privadas de AWS PrivateLink para los siguientes puntos de conexión:

- Puntos de conexión de bucket regionales (por ejemplo, `s3.us-east-1.amazonaws.com`)
- Puntos de conexión de control (por ejemplo, `s3-control.us-east-1.amazonaws.com`)
- Puntos de conexión de puntos de acceso (por ejemplo, `s3-accesspoint.us-east-1.amazonaws.com`)

Si tiene un punto de conexión de puerta de enlace en su VPC, puede dirigir automáticamente las solicitudes dentro de la VPC a través del punto de conexión de la puerta de enlace de S3 actual y las solicitudes en las instalaciones a través del punto de conexión de su interfaz. Este enfoque le permite optimizar sus costos de red mediante el uso de puntos de conexión de la puerta de enlace, que no se facturan, para el tráfico en la VPC. Sus aplicaciones en las instalaciones pueden utilizar AWS PrivateLink con la ayuda del punto de conexión de Resolver entrante. Amazon proporciona un servidor DNS, denominado Route 53 Resolver, para la VPC. Un punto de conexión de Resolver entrante reenvía las consultas de DNS desde la red local a Route 53 Resolver.

Important

Para aprovechar la ruta de red más económica al utilizar Habilitar el DNS privado solo para puntos de conexión entrantes, debe haber un punto de conexión de la puerta de enlace en la VPC. La presencia de un punto de conexión de puerta de enlace ayuda a garantizar que el tráfico en la VPC siempre se dirija a través de la red AWS privada cuando se selecciona la opción Habilitar el DNS privado solo para puntos de conexión entrantes. Debe mantener este punto de conexión de puerta de enlace mientras tenga seleccionada la opción Habilitar el DNS privado solo para puntos de conexión entrantes. Si desea eliminar el punto de conexión de puerta de enlace, primero debe desmarcar Habilitar el DNS privado solo para puntos de conexión entrantes.

Si desea actualizar un punto de conexión de interfaz existente para Habilitar el DNS privado solo para los puntos de conexión entrantes, primero debe confirmar que su VPC tenga un punto de conexión de puerta de enlace de S3. Para obtener más información sobre los puntos de conexión de la puerta de enlace y la administración de los nombres DNS privados, consulte los Puntos de conexión de VPC de la puerta de enlace y Administración de nombres de DNS, respectivamente, en la Guía de AWS PrivateLink.

La opción Habilitar el DNS privado solo para los puntos de conexión entrantes solo está disponible para los servicios que admiten puntos de conexión de la puerta de enlace.

Para obtener más información sobre la creación de un punto de conexión de VPC que utilice Habilitar el DNS privado solo para los puntos de conexión entrantes, consulte Crear un punto de conexión de la interfaz en la Guía de AWS PrivateLink.

Uso de la consola de la VPC

En la consola, tiene dos opciones: Habilitar el nombre de DNS y Habilitar el DNS privado solo para los puntos de conexión entrantes. Habilitar el nombre de DNS es una opción admitida por AWS PrivateLink. Al utilizar la opción Habilitar el nombre de DNS, puede utilizar la conectividad privada de Amazon con Amazon S3 y, al mismo tiempo, realizar solicitudes a los nombres de DNS de puntos de conexión públicos predeterminados. Cuando esta opción está habilitada, los clientes pueden aprovechar la ruta de red más económica disponible para su aplicación.

Al habilitar los nombres DNS privados en un punto de conexión de la interfaz de VPC nuevo o existente para Amazon S3, se selecciona la opción Habilitar el DNS privado solo para los puntos de conexión entrantes de forma predeterminada. Si se selecciona esta opción, las aplicaciones utilizan únicamente los puntos de conexión de la interfaz para el tráfico en las instalaciones. Este tráfico dentro de la VPC utiliza automáticamente los puntos de conexión de la puerta de enlace más económicos. También puede desmarcar la opción Habilitar el DNS privado solo para puntos de conexión entrantes para dirigir todas las solicitudes de S3 a través del punto de conexión de la interfaz.

Uso de AWS CLI

Si no especifica un valor para `PrivateDnsOnlyForInboundResolverEndpoint`, el valor predeterminado es `true`. Sin embargo, antes de que la VPC aplique la configuración, realiza una comprobación para asegurarse de que haya un punto de conexión de la puerta de enlace presente en la VPC. Si hay un punto de conexión de la puerta de enlace en la VPC, la llamada se realiza correctamente. De lo contrario, si ve el siguiente mensaje de error:

Para configurar PrivateDnsOnlyForInboundResolverEndpoint en true, la VPC *`vpce_id`* debe tener un punto de conexión de la puerta de enlace para el servicio.

Para un nuevo punto de conexión de la interfaz de la VPC

Utilice los atributos `private-dns-enabled` y `dns-options` para habilitar el DNS privado a través de la línea de comandos. La opción `PrivateDnsOnlyForInboundResolverEndpoint` del atributo `dns-options` debe estar establecida en `true`. Reemplace los *`user input placeholders`* con su propia información.

```
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
```

Para un punto de conexión de VPC existente

Si quiere usar un DNS privado para un punto de conexión de VPC existente, use el siguiente comando de ejemplo y reemplace *user input placeholders* con su información.

```
aws ec2 modify-vpc-endpoint \
--region us-east-1 \
--vpc-endpoint-id client-vpc-id \
--private-dns-enabled \
--dns-options PrivateDnsOnlyForInboundResolverEndpoint=false
```

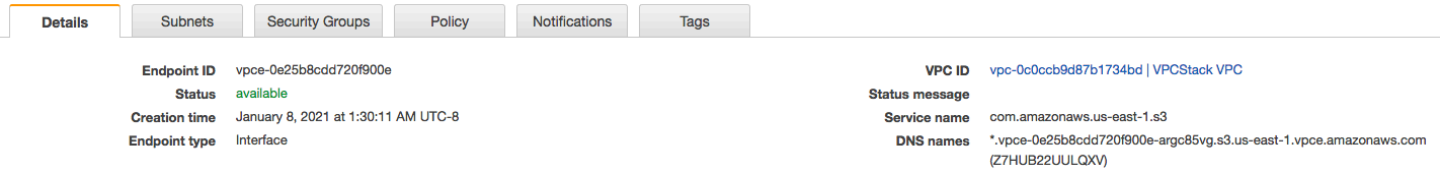
Si desea actualizar un punto de conexión de VPC existente para habilitar el DNS privado solo para el Resolver entrante, utilice el siguiente ejemplo y sustituya los valores de ejemplo por los suyos.

```
aws ec2 modify-vpc-endpoint \
--region us-east-1 \
--vpc-endpoint-id client-vpc-id \
--private-dns-enabled \
--dns-options PrivateDnsOnlyForInboundResolverEndpoint=true
```

Acceder a buckets, puntos de acceso y operaciones de la API de control de Amazon S3 desde los puntos de conexión de la interfaz de S3

Puede utilizar la AWS CLI o los SDK de AWS para acceder a los buckets, los puntos de acceso de S3 y las operaciones de la API de control de Amazon S3 a través de los puntos de conexión de la interfaz de S3.

La siguiente imagen muestra la pestaña Details (Detalles) de la consola de la VPC, donde puede encontrar el nombre DNS de un punto de conexión de la VPC. En este ejemplo, el ID de punto de conexión de la VPC (vpce-id) es `vpce-0e25b8cdd720f900e` y el nombre DNS es `*.vpce-0e25b8cdd720f900e-argc85vg.s3.us-east-1.vpce.amazonaws.com`.



Cuando utilice el nombre de DNS para acceder a un recurso, sustituya * por el valor correspondiente. Los valores adecuados que se deben utilizar en lugar de * son los siguientes:

- `bucket`
- `accesspoint`
- `control`

Por ejemplo, para acceder a un bucket, utilice un nombre de DNS como este:

```
bucket.vpce-0e25b8cdd720f900e-argc85vg.s3.us-east-1.vpce.amazonaws.com
```

Para ver ejemplos de cómo usar los nombres de DNS para acceder a los buckets, los puntos de acceso y las operaciones de la API de control de Amazon S3, consulte las siguientes secciones de Ejemplos de AWS CLI y Ejemplos del SDK de AWS.

Para obtener más información acerca de cómo ver los nombres de DNS específicos de los puntos de conexión, consulte Visualización de la configuración de los nombres de DNS privados de servicio de los puntos de conexión en la Guía del usuario de VPC.

Ejemplos de AWS CLI

Utilice los parámetros `--region` y `--endpoint-url` para acceder a los buckets de S3, los puntos de acceso S3 o las operaciones de la API de control de Amazon S3 a través de los puntos de conexión de la interfaz de S3 en los comandos AWS CLI.

Ejemplo: utilizar una URL del punto de conexión para enumerar objetos en su bucket

En el siguiente ejemplo, reemplace el nombre del bucket *my-bucket*, la región *us-east-1* y el nombre de DNS del ID de punto de conexión de VPC *vpce-1a2b3c4d-5e6f.s3.us-east-1.vpce.amazonaws.com* con su información.

```
aws s3 ls s3://my-bucket/ --region us-east-1 --endpoint-url
 https://bucket.vpce-1a2b3c4d-5e6f.s3.us-east-1.vpce.amazonaws.com
```

Ejemplo: utilizar una URL del punto de conexión para enumerar objetos de un punto de acceso

- Método 1: usar el nombre de recurso de Amazon (ARN) del punto de acceso con el punto de conexión del punto de acceso

  Reemplace el ARN *us-east-1:123456789012:accesspoint/accesspointexamplename*, la región *us-east-1* y el ID de punto de conexión de VPC *vpce-1a2b3c4d-5e6f.s3.us-east-1.vpce.amazonaws.com* con su información.

  ```
  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 no puede ejecutar el comando correctamente, actualice su AWS CLI a la versión más reciente e inténtelo de nuevo. Para obtener más información sobre la instalación de actualización, consulte Instalación o actualización de la versión más reciente de AWS CLI en la Guía del usuario de AWS Command Line Interface.

- Método 2: usar el alias del punto de acceso con el punto de conexión del bucket regional

  En el siguiente ejemplo, reemplace el alias del punto de acceso *accesspointexamplename-8tyekmigicmhun8n9kwpfur39dnw4use1a-s3alias*, la región *us-east-1* y el ID de punto de conexión de VPC *vpce-1a2b3c4d-5e6f.s3.us-east-1.vpce.amazonaws.com* con su información.

```
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
```

- Método 3: usar el alias del punto de acceso con el punto de conexión del punto de acceso

  En primer lugar, para crear un punto de conexión de S3 con el bucket incluido como parte del nombre de host, defina el estilo de direccionamiento hacia `virtual` para que lo utilice `aws s3api`. Para obtener más información sobre `AWS configure`, consulte Opciones de los archivos de configuración y credenciales en la Guía del usuario de AWS Command Line Interface.

  ```
  aws configure set default.s3.addressing_style virtual
  ```

  En el siguiente ejemplo, reemplace el alias del punto de acceso *accesspointexamplename-8tyekmigicmhun8n9kwpfur39dnw4use1a-s3alias*, la región *us-east-1* y el ID de punto de conexión de VPC *vpce-1a2b3c4d-5e6f.s3.us-east-1.vpce.amazonaws.com* con su información. Para obtener más información acerca de los alias de punto de acceso, consulte Uso de un alias de estilo de bucket para su punto de acceso de bucket de 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
  ```

Ejemplo: utilizar una URL del punto de conexión para enumerar trabajos con una operación de la API de control de S3

En el siguiente ejemplo, reemplace la región *us-east-1*, el ID de punto de conexión de VPC *vpce-1a2b3c4d-5e6f.s3.us-east-1.vpce.amazonaws.com* y el ID de cuenta *12345678* con su información.

```
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
```

Ejemplos del SDK de AWS

Para acceder a los buckets de S3, los puntos de acceso de S3 o las operaciones de la API de control de Amazon S3 a través de los puntos de conexión de la interfaz de S3 al utilizar los SDK de AWS, actualice sus SDK a la versión más actual. A continuación, configure sus clientes para que utilicen una URL de punto de conexión para acceder a un bucket, un punto de acceso o a operaciones de la API de control de Amazon S3 a través de los puntos de conexión de la interfaz de S3.

SDK for Python (Boto3)

Ejemplo: utilizar una URL de punto de conexión para acceder a un bucket de S3

En el siguiente ejemplo, reemplace la región *us-east-1* y el ID de punto de conexión de VPC *vpce-1a2b3c4d-5e6f.s3.us-east-1.vpce.amazonaws.com* con su información.

```
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'
)
```

Ejemplo: utilizar una URL de punto de conexión para acceder a un punto de acceso S3

En el siguiente ejemplo, reemplace la región *us-east-1* y el ID de punto de conexión de VPC *vpce-1a2b3c4d-5e6f.s3.us-east-1.vpce.amazonaws.com* con su información.

```
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'
)
```

Ejemplo: utilizar una URL de punto de conexión para acceder a la API de control S3

En el siguiente ejemplo, reemplace la región *us-east-1* y el ID de punto de conexión de VPC *vpce-1a2b3c4d-5e6f.s3.us-east-1.vpce.amazonaws.com* con su información.

```
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

Ejemplo: utilizar una URL de punto de conexión para acceder a un bucket de S3

En el siguiente ejemplo, reemplace el ID de punto de conexión de VPC *vpce-1a2b3c4d-5e6f.s3.us-east-1.vpce.amazonaws.com* con su información.

```
// 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();
```

Ejemplo: utilizar una URL de punto de conexión para acceder a un punto de acceso S3

En el siguiente ejemplo, reemplace el ID de punto de conexión de VPC *vpce-1a2b3c4d-5e6f.s3.us-east-1.vpce.amazonaws.com* y el ARN *us-east-1:123456789012:accesspoint/prod* con su información.

```
// 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");
```

Ejemplo: utilizar una URL de punto de conexión para acceder a una operación de la API de control de Amazon S3

En el siguiente ejemplo, reemplace el ID de punto de conexión de VPC *vpce-1a2b3c4d-5e6f.s3.us-east-1.vpce.amazonaws.com* con su información.

```
// 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

Ejemplo: utilizar una URL de punto de conexión para acceder a un bucket de S3

En el siguiente ejemplo, reemplace el ID de punto de conexión de VPC *vpce-1a2b3c4d-5e6f.s3.us-east-1.vpce.amazonaws.com* y la región *Region.US_EAST_1* con su información.

```
// 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()
```

Ejemplo: utilizar una URL de punto de conexión para acceder a un punto de acceso S3

En el siguiente ejemplo, reemplace el ID de punto de conexión de VPC *vpce-1a2b3c4d-5e6f.s3.us-east-1.vpce.amazonaws.com* y la región *Region.US_EAST_1* con su información.

```
// 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()
```

Ejemplo: utilizar una URL de punto de conexión para acceder a la API de control S3

En el siguiente ejemplo, reemplace el ID de punto de conexión de VPC *vpce-1a2b3c4d-5e6f.s3.us-east-1.vpce.amazonaws.com* y la región *Region.US_EAST_1* con su información.

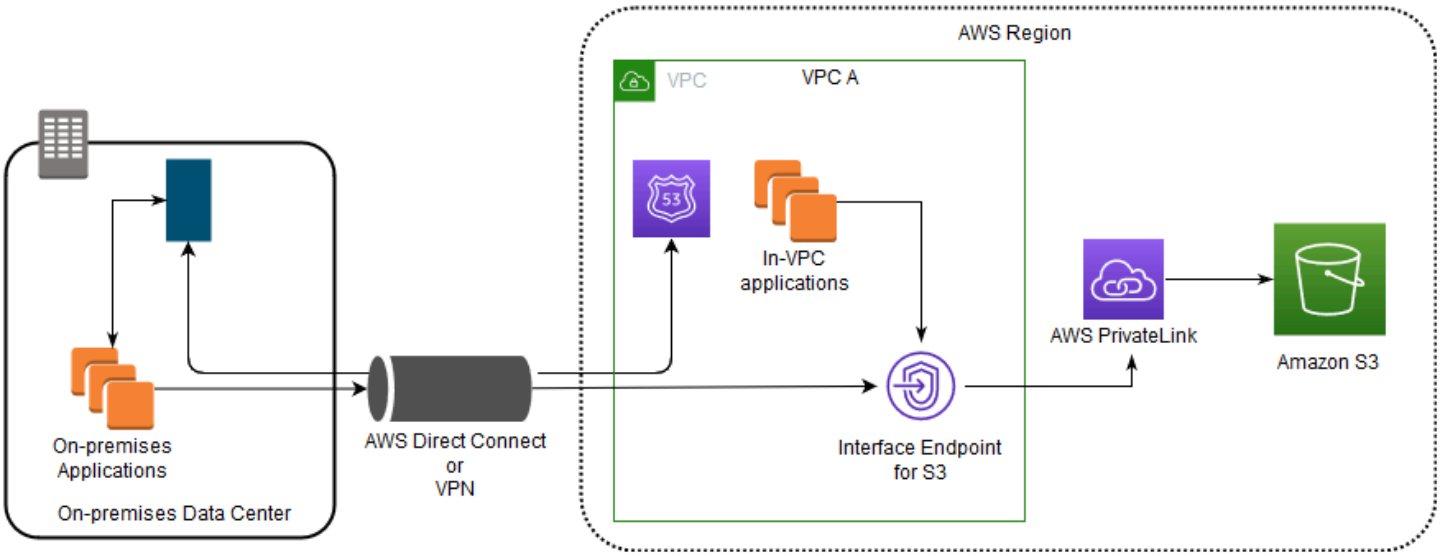
```
// control client
Region region = Region.US_EAST_1;
s3ControlClient = S3ControlClient.builder().region(region)

 .endpointOverride(URI.create("https://control.vpce-1a2b3c4d-5e6f.s3.us-
east-1.vpce.amazonaws.com"))
                                  .build()
```

Actualización de una configuración DNS en las instalaciones

Al utilizar nombres DNS específicos de puntos de enlace para acceder a los puntos de enlace de la interfaz de Amazon S3, no es necesario actualizar la resolución DNS local. Puede resolver el nombre DNS específico del punto de conexión con la dirección IP privada del punto de conexión de la interfaz desde el dominio DNS público de Amazon S3.

Uso de puntos de enlace de interfaz para acceder a Amazon S3 sin un punto de conexión de gateway o una gateway de Internet en la VPC

Los puntos de enlace de interfaz de su VPC pueden dirigir tanto las aplicaciones en VPC como las aplicaciones locales a Amazon S3 a través de la red de Amazon, como se muestra en el siguiente diagrama.

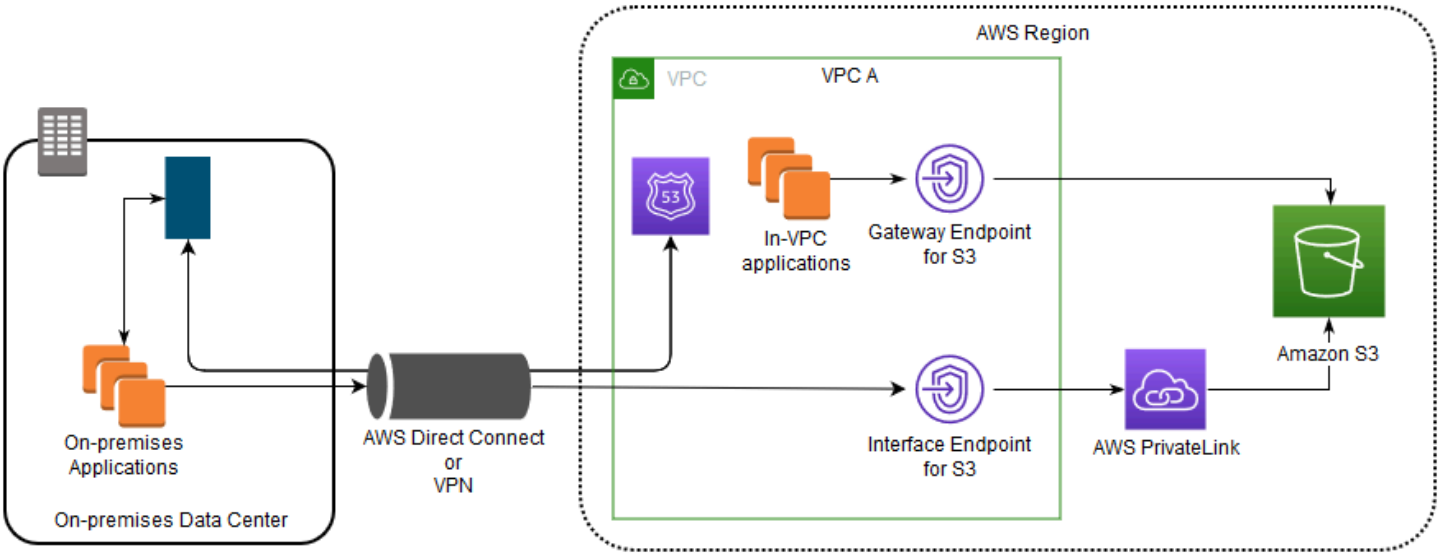


En el siguiente diagrama se ilustra lo siguiente:

- Su red en las instalaciones utiliza AWS Direct Connect o AWS VPN para conectarse a la VPC A.
- Las aplicaciones locales y en la VPC A utilizan nombres DNS específicos de punto de conexión para acceder a Amazon S3 a través del punto de conexión de la interfaz S3.
- Las aplicaciones en las instalaciones envían datos al punto de conexión de la interfaz en la VPC a través de AWS Direct Connect (o AWS VPN). AWS PrivateLink transfiere los datos desde el punto de conexión de la interfaz hasta Amazon S3 a través de la red de AWS.
- Las aplicaciones en la VPC también envían tráfico al punto de conexión de la interfaz. AWS PrivateLink transfiere los datos desde el punto de conexión de la interfaz a Amazon S3 a través de la red de AWS.

Uso de puntos de enlace de gateway y puntos de enlace de interfaz juntos en la misma VPC para acceder a Amazon S3

Puede crear puntos de enlace de interfaz y conservar el punto de conexión de gateway existente en la misma VPC, como se muestra en el siguiente diagrama. De este modo, permite que las aplicaciones de la VPC continúen accediendo a Amazon S3 a través del punto de conexión de la puerta de enlace, lo que no se factura. En ese caso, solo las aplicaciones en las instalaciones utilizarían puntos de conexión de la interfaz para acceder a Amazon S3. A fin de acceder a Amazon S3 de esta manera, debe actualizar las aplicaciones en las instalaciones para utilizar nombres de DNS específicos de puntos de conexión para Amazon S3.



En el siguiente diagrama se ilustra lo siguiente:

- Las aplicaciones en las instalaciones utilizan nombres de DNS específicos de cada punto de conexión para enviar datos al punto de conexión de la interfaz dentro de la VPC a través de AWS Direct Connect (o AWS VPN). AWS PrivateLink transfiere los datos desde el punto de conexión de la interfaz hasta Amazon S3 a través de la red de AWS.
- Mediante el uso de nombres regionales predeterminados de Amazon S3, las aplicaciones en VPC envían datos al punto de conexión de gateway que se conecta a Amazon S3 a través de la red de AWS.

Para obtener más información acerca de los puntos de enlace de gateway, consulte Puntos de enlace de la VPC de gateway en la Guía del usuario de VPC.

Creación de una política de puntos de enlace de la VPC para Amazon S3

Puede asociar una política de puntos de enlace con su punto de conexión de la VPC que controla el acceso a Amazon S3. La política especifica la siguiente información:

- La entidad principal de AWS Identity and Access Management (IAM) que puede realizar acciones
- Las acciones que se pueden realizar
- Los recursos en los que se pueden llevar a cabo las acciones

También puede utilizar las políticas de bucket de Amazon S3 para restringir el acceso a buckets específicos desde un punto de conexión de VPC específico utilizando la condición `aws:sourceVpce` de su política de bucket. En los siguientes ejemplos se muestran políticas que restringen el acceso a un bucket o a un punto de conexión.

Temas

- Ejemplo: restringir el acceso a un bucket específico desde un punto de conexión de la VPC
- Ejemplo: restringir el acceso a los buckets a una cuenta específica desde un punto de conexión de la VPC
- Ejemplo: restringir el acceso a un punto de conexión de la VPC específico en la política de bucket de S3

Ejemplo: restringir el acceso a un bucket específico desde un punto de conexión de la VPC

Puede crear una política de puntos de conexión que restrinja el acceso únicamente a buckets específicos de Amazon S3. Este tipo de política es útil si tiene otros Servicios de AWS en su VPC que utilicen buckets. La siguiente política de bucket restringe el acceso únicamente a *amzn-s3-demo-bucket1*. Para utilizar esta política de puntos de conexión, sustituya *amzn-s3-demo-bucket1* por el nombre de su bucket.

```
{
  "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:::amzn-s3-demo-bucket1",
                   "arn:aws:s3:::amzn-s3-demo-bucket1/*"]
    }
  ]
}
```

Ejemplo: restringir el acceso a los buckets a una cuenta específica desde un punto de conexión de la VPC

Puede crear una política de punto de conexión que restrinja el acceso solo a los buckets de S3 de una Cuenta de AWS específica. Para evitar que los clientes de su VPC accedan a los buckets que no posee, utilice la siguiente instrucción en su política de puntos de conexión. En el siguiente ejemplo de instrucción, se crea una política que restringe el acceso a los recursos pertenecientes a un único ID de Cuenta de 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**
>
> Puede utilizar la clave Cuenta de AWS o `aws:ResourceAccount` en la política de IAM para especificar el ID de `s3:ResourceAccount` del recurso al que se va a acceder. Sin embargo, debe tener en cuenta que algunos Servicios de AWS dependen del acceso a buckets administrados por AWS. Por lo tanto, es posible que el uso de la clave `aws:ResourceAccount` o `s3:ResourceAccount` de la política de IAM también afecte al acceso a estos recursos.

Ejemplo: restringir el acceso a un punto de conexión de la VPC específico en la política de bucket de S3

Ejemplo: restringir el acceso a un punto de conexión de la VPC específico en la política de bucket de S3

La siguiente política de bucket de Amazon S3 permite el acceso a un bucket específico, *amzn-s3-demo-bucket2*, solo desde un punto de conexión de VPC *vpce-1a2b3c4d*. La política deniega todo el acceso al bucket si el punto de conexión especificado no se está utilizando. La condición `aws:sourceVpce` especifica el punto de conexión y no requiere un nombre de recurso de Amazon (ARN) para el recurso de punto de conexión de VPC, solo el ID de punto de conexión. Para utilizar esta política de buckets, sustituya *amzn-s3-demo-bucket2* y *vpce-1a2b3c4d* por el nombre del bucket y el punto de conexión.

Important

- Al aplicar las siguientes políticas de bucket de Amazon S3 para restringir el acceso únicamente a determinados puntos de conexión de VPC, puede bloquear el acceso al bucket de forma inintencionada. Las políticas de bucket pensadas para limitar específicamente el acceso del bucket a las conexiones procedentes de su punto de conexión de VPC pueden bloquear todas las conexiones al bucket. Para obtener información acerca de cómo corregir este problema, consulte Mi política de bucket tiene una VPC o un ID de punto de conexión de la VPC incorrectos. ¿Cómo puedo corregir la política de modo que pueda tener acceso al bucket? en el Centro de conocimientos de AWS Support.
- Antes de utilizar la política de ejemplo siguiente, reemplace el ID del punto de conexión de la VPC por un valor adecuado para su caso de uso. De lo contrario, no podrá acceder a su bucket.
- Esta política deshabilita el acceso a la consola al bucket especificado, ya que las solicitudes de consola no se originan en el punto de conexión de la VPC especificado.

```
{
  "Version": "2012-10-17",
  "Id": "Policy1415115909152",
  "Statement": [
    { "Sid": "Access-to-specific-VPCE-only",
      "Principal": "*",
      "Action": "s3:*",
      "Effect": "Deny",
      "Resource": ["arn:aws:s3:::amzn-s3-demo-bucket2",
                   "arn:aws:s3:::amzn-s3-demo-bucket2/*"],
      "Condition": {"StringNotEquals": {"aws:sourceVpce": "vpce-1a2b3c4d"}}
    }
  ]
}
```

Para obtener más ejemplos de políticas, consulte Puntos de enlace para Amazon S3 en la Guía del usuario de VPC.

Para obtener más información acerca de la conectividad de VPC, consulte Opciones de conectividad de red a VPC en el documento técnico de AWSOpciones de conectividad de Amazon Virtual Private Cloud.

Administración de accesos

En AWS, un recurso es una entidad con la que se puede trabajar. En Amazon Simple Storage Service (S3), los buckets y los objetos son los recursos originales de Amazon S3. Es probable que todos los clientes de S3 tengan buckets con objetos en ellos. A medida que se agregaban nuevas funciones a S3, también se agregaban recursos adicionales, pero no todos los clientes utilizan estos recursos específicos de características. Para obtener más información sobre los recursos de Amazon S3, consulte Recursos de S3.

De forma predeterminada, todos los recursos de Amazon S3 son privados. De forma predeterminada, el usuario raíz de la Cuenta de AWS que creó el recurso (propietario del recurso) y los usuarios de IAM de esa cuenta con los permisos necesarios pueden acceder al recurso que hayan creado. El propietario del recurso decide quién más puede acceder al recurso y las acciones que otros pueden realizar en el recurso. S3 cuenta con varias herramientas de administración de acceso que puede utilizar para conceder a otras personas el acceso a sus recursos de S3.

En las siguientes secciones, encontrará información general sobre los recursos de S3, las herramientas de administración de acceso de S3 disponibles y los mejores casos de uso de cada herramienta de administración de acceso. Las listas de estas secciones pretenden ser exhaustivas e incluir todos los recursos de S3, las herramientas de administración de acceso y los casos de uso comunes de la administración de acceso. Al mismo tiempo, estas secciones están diseñadas para ser directorios que lo guíen a los detalles técnicos que desee. Si conoce bien algunos de los siguientes temas, puede pasar directamente a la sección que se aplique a su caso.

Temas

Recursos de S3

Los recursos originales de Amazon S3 son buckets y los objetos que contienen. A medida que se agregaban nuevas características a S3, también se agregaban nuevos recursos. A continuación se muestra una lista completa de recursos de S3 y sus características respectivas.

Tipo de recurso	Característica de Amazon S3	Descripción
`bucket`	Características principales	Un bucket es un contenedor de objetos. Para almacenar un objeto en S3, cree un bucket y, a continuación, cargue uno o más objetos en el bucket. Para obtener más información, consulte Creación, configuración y trabajo con buckets de Amazon S3.
`object`		Un objeto puede ser un archivo y cualquier metadato que describa ese archivo. Cuando un objeto está en el bucket, puede abrirlo, descargarlo y moverlo. Para obtener más información, consulte Cargar, descargar y trabajar con objetos en Amazon S3.
`accesspoint`	Puntos de acceso	Los puntos de acceso son puntos de conexión de red con nombre y asociados a los buckets que se pueden utilizar para realizar operaciones con objetos de Amazon S3, como `GetObject` y `PutObject` . Cada punto de acceso tiene permisos distintos, controles de red y una política de punto de acceso personalizada que funciona en conjunción con la política de bucket asociada al bucket subyacente. Puede configurar cualquier punto de acceso para que acepte solo las solicitudes procedentes de una nube privada virtual (VPC) o configurar ajustes personalizados de bloquear acceso público para cada punto de acceso. Para obtener más información, consulte Administración del acceso a datos con puntos de acceso de Amazon S3.
`objectlambdaaccesspoint`		Un punto de acceso de Object Lambda es un punto de acceso para un bucket que también está asociado a una función de Lambda. Con el punto de acceso de Object

Tipo de recurso	Característica de Amazon S3	Descripción
		Lambda, puede agregar su propio código a las solicitudes `GET`, `LIST` y `HEAD` de Amazon S3 para modificar y procesar los datos a medida que vuelven a una aplicación. Para obtener más información, consulte Creación de puntos de acceso Object Lambda.
`multiregionaccesspoint`		Los puntos de acceso multirregión proporcionan un punto de conexión global que las aplicaciones pueden utilizar para satisfacer las solicitudes de los buckets de Amazon S3 ubicados en varias regiones de AWS. Puede utilizar puntos de acceso multirregional para crear aplicaciones multirregionales con la misma arquitectura que se utiliza en una sola región y, a continuación, ejecutar esas aplicaciones en cualquier parte del mundo. En lugar de enviar las solicitudes a través de la congestionada Internet pública, las solicitudes de aplicación realizadas a un punto de conexión global de punto de acceso multirregión se dirigen automáticamente a través de la red global de AWS hacia el bucket de Amazon S3 más próximo. Para obtener más información, consulte Puntos de acceso de varias regiones de Amazon S3.
`job`	Operaciones por lotes de S3	Un trabajo es un recurso de la característica de Operaciones por lotes de S3. Puede utilizar Operaciones por lotes de S3 para realizar operaciones por lotes a gran escala en listas de objetos de Amazon S3 que especifique. Amazon S3 realiza un seguimiento del avance del trabajo de operación por lotes, envía notificaciones y guarda un informe de finalización de todas las acciones, por lo que proporciona una experiencia sin servidor, auditable y completamente administrada. Para obtener más información, consulte Realización de operaciones por lotes a gran escala en objetos de Amazon S3.

Tipo de recurso	Característica de Amazon S3	Descripción
`storagelensconfiguration`	Almacenamiento de lente de S3	Una configuración de Lente de almacenamiento de S3 recopila métricas de almacenamiento en toda la organización y datos de usuario entre cuentas. La Lente de almacenamiento de S3 proporciona a los administradores una visión única del uso y la actividad del almacenamiento de objetos en cientos o incluso miles de cuentas de una organización, con detalles para generar información en varios niveles de agregación. Para obtener más información, consulte Evaluación de la actividad y el uso de almacenamiento con Amazon S3 Storage Lens.
`storagelensgroup`		Un grupo de Lente de almacenamiento de S3 Storage Lens agrega métricas mediante filtros personalizados en función de metadatos de objetos. Los grupos de Lente de almacenamiento de S3 le ayudan a investigar las características de sus datos, como la distribución de los objetos por antigüedad, sus tipos de archivos más comunes, etc. Para obtener más información, consulte Trabajo con grupos de S3 Storage Lens.
`accessgrantsinstance`	Permisos de acceso de S3	Una instancia de Concesiones de acceso a S3 es un contenedor para las concesiones de S3 que cree. Con Concesiones de acceso a S3, puede crear concesiones a sus datos de Amazon S3 para las identidades de IAM de su cuenta, las identidades de IAM de otras cuentas (entre cuentas) y las identidades de directorio añadidas a AWS IAM Identity Center desde su directorio corporativo. Para obtener más información sobre Concesiones de acceso a S3, consulte Administración del acceso con S3 Access Grants.

Tipo de recurso	Característica de Amazon S3	Descripción
`accessgrantslocation`		Una ubicación de Concesiones de acceso es un bucket, un prefijo dentro de un bucket o un objeto que se registra en la instancia de Concesiones de acceso de S3. Debe registrar las ubicaciones en la instancia de Concesiones de acceso de S3 antes de poder crear una concesión para esa ubicación. A continuación, con Concesiones de acceso a S3 puede conceder acceso al bucket, al prefijo o al objeto para las identidades de IAM de su cuenta, las identidades de IAM de otras cuentas (entre cuentas) y las identidades de directorio agregadas a AWS IAM Identity Center desde su directorio corporativo. Para obtener más información sobre Concesiones de acceso a S3, consulte Administración del acceso con S3 Access Grants
`accessgrant`		Una concesión de acceso es una concesión individual a sus datos de Amazon S3. Con Concesiones de acceso a S3, puede crear concesiones a sus datos de Amazon S3 para las identidades de IAM de su cuenta, las identidades de IAM de otras cuentas (entre cuentas) y las identidades de directorio añadidas a AWS IAM Identity Center desde su directorio corporativo. Para obtener más información sobre Concesiones de acceso a S3, consulte Administración del acceso con S3 Access Grants

Buckets

Existen dos tipos de buckets de Amazon S3: buckets de uso general y buckets de directorio.

- Los bucket de uso general son del tipo de bucket original de S3 y se recomiendan para la mayoría de los casos de uso y patrones de acceso. Los bucket de uso general también permiten almacenar objetos en todas las clases de almacenamiento, excepto en S3 Express One Zone. Para obtener más información acerca de las clases de almacenamiento de S3, consulte Uso de las clases de almacenamiento de Amazon S3.

- Los buckets de directorio utilizan la clase de almacenamiento S3 Express One Zone, que se recomienda si la aplicación es sensible al rendimiento y se beneficia de latencias PUT y GET de un solo dígito de milisegundos. Para obtener más información, consulte Buckets de directorio, ¿Qué es S3 Express One Zone? y AWS Identity and Access Management (IAM) para S3 Express One Zone.

Categorización de los recursos de S3

Amazon S3 ofrece características para categorizar y organizar los recursos de S3. La categorización de los recursos no solo es útil para organizarlos, sino que también permite establecer reglas de administración de acceso en función de las categorías de recursos. En concreto, los prefijos y el etiquetado son dos características de organización del almacenamiento que puede utilizar al configurar los permisos de administración de acceso.

> **Note**
>
> La siguiente información se aplica a los buckets de uso general. Los buckets de directorio no admiten el etiquetado y tienen limitaciones de prefijos. Para obtener más información, consulte AWS Identity and Access Management (IAM) para S3 Express One Zone.

- Prefijos: un prefijo en Amazon S3 es una cadena de caracteres al principio del nombre de una clave de objeto que se utiliza para organizar los objetos en sus bucket de S3. Puede usar un carácter delimitador, como una barra inclinada (/), para indicar el final del prefijo dentro del nombre de la clave del objeto. Por ejemplo, puede tener nombres de clave de objeto que comiencen por el prefijo `engineering/` o nombres de clave de objeto que comiencen por el prefijo `marketing/campaigns/`. El uso de un delimitador al final del prefijo, como una barra inclinada `/`, emula las convenciones de nomenclatura de carpetas y archivos. Sin embargo, en S3, el prefijo forma parte del nombre de la clave de objeto. En los buckets de S3 de uso general, no existe una jerarquía de carpetas real.

  Amazon S3 permite organizar y agrupar objetos mediante sus prefijos. También puede administrar el acceso a los objetos mediante sus prefijos. Por ejemplo, puede limitar el acceso a solo los objetos con nombres que comiencen con un prefijo concreto.

  Para obtener más información, consulte Organizar objetos con prefijos. La consola S3 utiliza el concepto de carpetas que, en los buckets de uso general, son básicamente prefijos que se

anteponen al nombre de la clave del objeto. Para obtener más información, consulte Organización de objetos en la consola de Amazon S3 con carpetas.

- Etiquetas: cada etiqueta es un par de clave-valor que puede asignar a los recursos. Por ejemplo, puede etiquetar algunos recursos con la etiqueta `topicCategory=engineering`. Puede utilizar el etiquetado para facilitar la asignación de costos, la categorización, la organización y el control de acceso. El etiquetado de buckets solo se utiliza para la asignación de costos. Puede etiquetar objetos, Lente de almacenamiento de S3, trabajos y Concesiones de acceso a S3 con fines de organización o control de acceso. En Concesiones de acceso a S3, también puede utilizar el etiquetado para asignar los costos. Como ejemplo del control del acceso a los recursos mediante sus etiquetas, puede compartir solo los objetos que tengan una etiqueta o una combinación de etiquetas concretas.

  Para obtener más información, consulte Control del acceso a los recursos de AWS mediante etiquetas de recursos en la Guía del usuario de IAM.

Identidades

En Amazon S3, el propietario del recurso es la identidad que creó el recurso, como un bucket o un objeto. De forma predeterminada, solo el usuario raíz de la cuenta que creó el recurso y las identidades de IAM de la cuenta que tiene el permiso necesario puede acceder al recurso de S3. Los propietarios de los recursos pueden conceder a otras identidades el acceso a sus recursos de S3.

Las identidades que no posean un recurso pueden solicitar acceso a ese recurso. Todas las solicitudes a un recurso son autenticadas o no autenticadas. Las solicitudes autenticadas deben incluir un valor de firma que autentique al remitente de la solicitud, pero las solicitudes no autenticadas no requieren una firma. Le recomendamos que conceda acceso solo a los usuarios autenticados. Para obtener más información acerca de la autenticación de solicitudes, consulte Realizar solicitudes.

> **Important**
>
> Recomendamos que no utilice las credenciales del usuario raíz de la Cuenta de AWS para realizar solicitudes autenticadas. En su lugar, cree un rol de IAM y concédale derechos de acceso completos. Los usuarios con este rol se denominan usuarios administradores. Puede utilizar las credenciales asignadas al rol de administrador, en lugar de las credenciales de usuario raíz de la Cuenta de AWS, para interactuar con AWS y realizar tareas, como crear un bucket, crear usuarios y conceder permisos. Para obtener más información, consulte

Credenciales de usuario raíz de Cuenta de AWS y credenciales de usuario de IAM en la Referencia general de AWS y las Prácticas recomendadas de seguridad en IAM en la Guía del usuario de IAM.

Las identidades que acceden a sus datos en Amazon S3 pueden ser una de las siguientes:

Propietario de la Cuenta de AWS

La Cuenta de AWS que creó el recurso. Por ejemplo, la cuenta que creó el bucket. Esta cuenta es propietaria del recurso. Para obtener más información, consulte Usuario raíz de la cuenta de AWS.

Identidades de IAM en la misma cuenta del propietario de la Cuenta de AWS

Al configurar cuentas para nuevos miembros del equipo que requieren acceso a S3, el propietario de la Cuenta de AWS puede usar AWS Identity and Access Management (IAM) para crear usuarios, grupos y roles. A continuación, el propietario de la Cuenta de AWS puede compartir recursos con estas identidades de IAM. El propietario de la cuenta también puede especificar los permisos para conceder las identidades de IAM, que permiten o deniegan las acciones que se pueden realizar en los recursos compartidos.

Las identidades de IAM proporcionan capacidades mejoradas, incluida la capacidad de requerir a los usuarios que especifiquen credenciales de inicio de sesión antes de acceder a recursos compartidos. Mediante el uso de identidades IAM puede implementar una forma de autenticación multifactor (MFA) de IAM para permitir una base de identidad sólida. Una práctica recomendada de IAM consiste en crear roles para la administración del acceso en lugar de conceder permisos a cada usuario individual. Se asigna a los usuarios individuales el rol adecuado. Para obtener más información, consulte prácticas recomendadas de seguridad en IAM.

Otros propietarios de cuentas de AWS y sus identidades de IAM (acceso entre cuentas)

El propietario de la Cuenta de AWS también puede dar acceso a los recursos a otros propietarios de cuentas de AWS o a identidades de IAM que pertenezcan a otra cuenta de AWS.

> **Note**
>
> Delegación de permisos: si una Cuenta de AWS es propietaria de un recurso, puede conceder esos permisos a otra Cuenta de AWS. Esa cuenta, a continuación, puede delegar esos permisos, o un subconjunto de ellos, a usuarios de la misma cuenta. Esto se conoce

como delegación de permisos. Sin embargo, una cuenta que recibe permisos de otra cuenta no puede delegar esos permisos “entre cuentas” a otra Cuenta de AWS.

Usuarios anónimos (acceso público)

El propietario de la Cuenta de AWS puede hacer públicos los recursos. Al hacer público un recurso, técnicamente se comparte el recurso con el usuario anónimo. Los buckets creados a partir de abril de 2023 bloquean todo el acceso público de forma predeterminada, a menos que cambie este ajuste. Le recomendamos que configure sus buckets para que bloqueen el acceso público y que solo conceda acceso a usuarios autenticados. Para obtener más información acerca del bloqueo del acceso público, consulte Bloquear el acceso público a su almacenamiento de Amazon S3.

Servicios de AWS

El propietario del recurso puede conceder a otro servicio de AWS acceso a un recurso de Amazon S3. Por ejemplo, puede conceder el permiso `s3:PutObject` del servicio de AWS CloudTrail para escribir archivos de registro en su bucket. Para obtener más información, consulte Proporcionar acceso a un servicio de AWS.

Identidades de directorios de empresa

El propietario del recurso puede conceder a los usuarios o roles de su directorio corporativo acceso a un recurso de S3 mediante Concesiones de acceso a S3. Para obtener más información sobre cómo agregar el directorio corporativo a AWS IAM Identity Center, consulte What Is IAM Identity Center?.

Propietarios de buckets o recursos

La Cuenta de AWS que se usa para crear buckets y cargar objetos es la propietaria de dichos recursos. Un propietario del bucket puede conceder permisos de una cuenta a otra Cuenta de AWS (o a usuarios de otra cuenta) para cargar objetos.

Cuando el propietario de un bucket permite que otra cuenta cargue objetos a un bucket, el propietario del bucket, de forma predeterminada, es el propietario de todos los objetos cargados en su bucket. Sin embargo, si están desactivados tanto los ajustes de bucket de Aplicada al propietario del bucket como de Propietario del bucket preferido, la Cuenta de AWS que carga los objetos es la propietaria de esos objetos y el propietario del bucket no tiene permisos sobre los objetos que son propiedad de otra cuenta, con las siguientes excepciones:

- El propietario del bucket es quien paga las facturas. El propietario del bucket puede denegar el acceso a cualquier objeto, o eliminar cualquier objeto del bucket, independientemente de quién sea el propietario.
- El propietario del bucket puede archivar los objetos o restaurar los objetos archivados, independientemente de quién sea el propietario. El archivado se refiere a la clase de almacenamiento empleada para almacenar los objetos. Para obtener más información, consulte Administración del ciclo de vida del almacenamiento.

Herramientas de administración de acceso

Amazon S3 proporciona una variedad de características y herramientas de seguridad. A continuación se muestra una lista completa de estas características y herramientas. No necesita todas estas herramientas de administración de acceso, pero debe usar una o más para conceder acceso a sus recursos de Amazon S3. La aplicación adecuada de estas herramientas puede ayudarle a garantizar que solo pueden acceder a sus recursos los usuarios previstos.

La herramienta de administración de acceso más utilizada es una política de acceso. Una política de acceso puede ser una política basada en recursos que se asocia a un recurso de AWS, como una política de bucket para un bucket. Una política de acceso también puede ser una política basada en identidades que se asocia a una identidad AWS Identity and Access Management (IAM), como un usuario, grupo o rol de IAM. Escriba una política de acceso para conceder a Cuentas de AWS y usuarios, grupos y roles de IAM permisos para realizar operaciones en un recurso. Por ejemplo, puede conceder un permiso `PUT Object` a otra cuenta Cuenta de AWS de modo que la otra cuenta pueda cargar objetos en su bucket.

Una política de acceso describe quién tiene acceso a qué elemento. Cuando Amazon S3 recibe una solicitud, debe evaluar todas las políticas de acceso para determinar si debe autorizar o denegar la solicitud. Para obtener más información sobre evalúa estas políticas Amazon S3, consulte Cómo autoriza Amazon S3 una solicitud.

A continuación se muestran las herramientas de administración de acceso disponibles en Amazon S3.

Política de bucket

Una política de bucket de Amazon S3 es una política basada en recursos de AWS Identity and Access Management (IAM) con formato JSON que se asocia a un bucket concreto. Utilice políticas de bucket para conceder a otras Cuentas de AWS o a identidades de IAM permisos para el bucket

y los objetos que contiene. Muchos casos de uso de administración de acceso de S3 se pueden cumplir con una política de bucket. Con las políticas de bucket, puede personalizar el acceso al bucket para garantizar que solo las identidades que haya aprobado puedan acceder a los recursos y realizar acciones dentro de ellos. Para obtener más información, consulte Políticas de buckets para Amazon S3.

A continuación se muestra un ejemplo de política de bucket. La política de bucket se expresa con un archivo JSON. Este ejemplo de política concede a un rol de IAM permiso de lectura para todos los objetos en el bucket. Contiene una instrucción llamada `BucketLevelReadPermissions`, que permite la acción `s3:GetObject` (permiso de lectura) en objetos que estén en un bucket llamado `amzn-s3-demo-bucket1`. Al especificar un rol de IAM como `Principal`, esta política concede acceso a cualquier usuario de IAM con este rol. Para utilizar esta política de ejemplo, sustituya *`user input placeholders`* por su propia información.

```
{
  "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:::amzn-s3-demo-bucket1/*"]
    }]
  }
```

> **Note**
>
> Al crear políticas, evite el uso de caracteres comodín (*) en el elemento `Principal` porque al usarlo permite a cualquiera acceder a sus recursos de Amazon S3. En lugar de ello, enumere explícitamente los usuarios o grupos a los que se les permite acceder al bucket, o enumere las condiciones que deben cumplirse mediante una cláusula de condición en la política. Además, en lugar de incluir un carácter comodín para las acciones de los usuarios o grupos, concédales permisos concretos cuando corresponda.

Políticas basadas en identidad

Una política de usuario de IAM o basada en identidades es un tipo de política de AWS Identity and Access Management (IAM). Una política basada en identidades es una política con formato JSON que se asocia a los usuarios, grupos o roles de IAM de su cuenta de AWS. Puede utilizar políticas basadas en identidades para conceder a una identidad de IAM acceso a sus buckets u objetos. Puede crear usuarios, grupos y roles de IAM en su cuenta y asociarles políticas de acceso. A continuación, puede conceder acceso a los recursos de AWS, incluidos los recursos de Amazon S3. Para obtener más información, consulte Políticas basadas en identidad para Amazon S3.

A continuación se muestra un ejemplo de una política basada en identidades. La política de ejemplo permite al rol de IAM asociado que realice seis acciones de Amazon S3 (permisos) diferentes en un bucket y los objetos que contiene. Si asocia esta política a un rol de IAM en su cuenta y asigna el rol a algunos de sus usuarios de IAM, los usuarios con este rol podrán realizar estas acciones en los recursos (buckets) especificados en su política. Para utilizar esta política de ejemplo, sustituya *user input placeholders* por su propia información.

```
{
  "Version": "2012-10-17",
  "Statement": [
  {
    "Sid": "AssignARoleActions",
    "Effect": "Allow",
    "Action": [
   "s3:PutObject",
   "s3:GetObject",
   "s3:ListBucket",
   "s3:DeleteObject",
   "s3:GetBucketLocation"
    ],
    "Resource": [
   "arn:aws:s3:::amzn-s3-demo-bucket1/*",
   "arn:aws:s3:::amzn-s3-demo-bucket1"
    ]
    },
  {
    "Sid": "AssignARoleActions2",
    "Effect": "Allow",
    "Action": "s3:ListAllMyBuckets",
    "Resource": "*"
  }
  ]
```

```
}
```

Permisos de acceso de S3

Utilice Concesiones de acceso a S3 para crear concesiones de acceso a sus datos de Amazon S3 tanto para las identidades en los directorios de identidades corporativas, como Active Directory, como para las identidades AWS Identity and Access Management (de IAM). Concesiones de acceso a S3 le ayuda a administrar los permisos de datos a escala. Además, Concesiones de acceso a S3 registra la identidad del usuario final y la aplicación utilizada para acceder a los datos de S3 en AWS CloudTrail. Esto proporciona un historial de auditoría detallado que incluye hasta la identidad del usuario final para todos los accesos a los datos de sus buckets de S3. Para obtener más información, consulte Administración del acceso con S3 Access Grants.

Puntos de acceso

Los puntos de acceso de Amazon S3 simplifican la administración del acceso a los datos a escala para aplicaciones que utilizan conjuntos de datos compartidos en Amazon S3. Los puntos de acceso son puntos de conexión de red con nombre que están asociados a los buckets. Puede usar puntos de acceso para realizar operaciones de objetos de S3 a escala, como la carga y la recuperación de objetos. Un bucket puede tener hasta 10 000 puntos de acceso asociados y para cada punto de acceso puede aplicar permisos y controles de red diferenciados para proporcionarle un control detallado sobre el acceso a los objetos de S3. Los puntos de acceso de Amazon S3 se pueden asociar a buckets en la misma cuenta o en otra cuenta de confianza. Las políticas de puntos de acceso son políticas basadas en recursos que se evalúan junto con la política de bucket subyacente. Para obtener más información, consulte Administración del acceso a datos con puntos de acceso de Amazon S3.

Lista de control de acceso (ACL)

Una ACL es una lista de concesiones en la que se identifica el beneficiario y el permiso concedido. Puede utilizar las ACL para conceder permisos de lectura o escritura básicos a otras Cuentas de AWS. Las ACL usan un esquema XML específico de Amazon S3. Una ACL es un tipo de política de AWS Identity and Access Management (IAM). Una ACL de objeto se usa para administrar el acceso a un objeto y una ACL de bucket se usa para administrar el acceso a un bucket. Con las políticas de bucket, hay una sola política de bucket para la totalidad del bucket, pero las ACL de objetos se especifican por cada objeto. Le recomendamos que mantenga las ACL desactivadas, excepto en circunstancias inusuales en las que necesite controlar el acceso a cada objeto de manera individual. Para obtener más información sobre cómo utilizar las ACL, consulte Control de la propiedad de los objetos y desactivación de las ACL del bucket.

> **Warning**
>
> La mayoría de los casos de uso modernos de Amazon S3 ya no requieren el uso de ACL.

A continuación se muestra una ACL de bucket de ejemplo. La concesión en la ACL muestra un propietario del bucket que tiene permisos de control pleno.

```
<?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>
```

Propiedad del objeto

Para administrar el acceso a sus objetos, debe ser el propietario del objeto. Puede utilizar el ajuste en el nivel de bucket de Propiedad de objetos para controlar la propiedad de los objetos cargados en el bucket. Además, use Propiedad de objetos para activar las ACL. De forma predeterminada, la Propiedad de objetos se establece en el ajuste Aplicada al propietario del bucket y todas las ACL están deshabilitadas. Cuando las ACL están deshabilitadas, el propietario del bucket posee todos los objetos del bucket y administra el acceso a los datos de forma exclusiva. Para administrar el acceso, el propietario del bucket usa políticas u otra herramienta de administración de acceso, excluyendo las ACL. Para obtener más información, consulte Control de la propiedad de los objetos y desactivación de las ACL del bucket.

La Propiedad de objetos tiene tres ajustes que puede utilizar para controlar la propiedad de los objetos que se cargan en el bucket y activar las ACL:

ACL desactivadas

- Aplicada al propietario del bucket (predeterminado): las ACL están desactivadas y el propietario del bucket tiene automáticamente la propiedad y el control total sobre cada objeto del bucket. Las ACL no afectan a los permisos de los datos del bucket de S3. El bucket utiliza políticas exclusivamente para definir el control de acceso.

ACL activadas

- Propietario del bucket preferido: el propietario del bucket tiene la propiedad y el control total sobre los nuevos objetos que otras cuentas escriben en el bucket con la ACL predefinida `bucket-owner-full-control`.
- Escritor del objeto: la Cuenta de AWS que carga un objeto es propietaria del objeto, tiene control total sobre él y puede conceder a otros usuarios acceso a él a través de ACL.

Prácticas recomendadas adicionales

Plantéese el uso de los siguientes ajustes y herramientas de bucket para ayudar a proteger los datos en tránsito y en reposo, lo cual es crucial para mantener la integridad y la accesibilidad de los datos:

- Bloquear el acceso público: no desactive la configuración de nivel de bucket predeterminada Bloquear el acceso público. Este ajuste bloquea el acceso público a los datos de forma predeterminada. Para obtener más información acerca del bloqueo del acceso público, consulte Bloquear el acceso público a su almacenamiento de Amazon S3.
- Control de versiones de S3: para garantizar la integridad de los datos, puede implementar el ajuste del bucket de control versiones de S3, que versiona los objetos a medida que realiza actualizaciones, en lugar de sobrescribirlos. Puede usar el control de versiones de S3 para conservar, recuperar y restaurar una versión anterior, si es necesario. Para obtener más información sobre el control de versiones de S3, consulte Usar el control de versiones en buckets de S3.
- Bloqueo de objetos de S3: el bloqueo de objetos de S3 es otro ajuste que puede implementar para lograr la integridad de los datos. Esta característica puede implementar un modelo de escritura única y lectura múltiple (WORM) para almacenar objetos de forma inmutable. Para obtener más información acerca del Bloqueo de objetos, consulte Usar Bloqueo de objetos de S3.
- Cifrado de objetos: Amazon S3 ofrece varias opciones de cifrado de objetos que protegen los datos en tránsito y en reposo. El cifrado en el servidor cifra el objeto antes de guardarlo en discos

de sus centros de datos y, a continuación, lo descifra al descargar los objetos. Si autentica su solicitud y tiene permisos de acceso, no existe diferencia alguna en la forma de obtener acceso a objetos cifrados o sin cifrar. Para obtener más información, consulte Protección de los datos con el cifrado del servidor. S3 cifra los objetos recién cargados de forma predeterminada. Para obtener más información, consulte Establecer el comportamiento del cifrado predeterminado del lado del servidor para los buckets de Amazon S3. El cifrado del lado del cliente es el acto de cifrar datos antes de enviarlos a Amazon S3. Para obtener más información, consulte Protección de los datos con el cifrado del cliente.

- Métodos de firma Signature Version 4 es el proceso de agregar información de autenticación a las solicitudes de AWS enviadas por HTTP. Por seguridad, la mayoría de las solicitudes de AWS se firman con una clave de acceso, que se compone de un ID de clave de acceso y una clave de acceso secreta. Estas dos claves comúnmente se denominan credenciales de seguridad. Para obtener más información, consulte Autenticación de solicitudes (AWS Signature Version 4) y Proceso de firma de Signature Version 4.

Acciones

Para obtener una lista completa de las claves de condición y los permisos de S3, consulte Actions, resources, and condition keys for Amazon S3 en la Referencia de autorizaciones de servicios.

Acciones

Las acciones de AWS Identity and Access Management (IAM) de Amazon S3 son las posibles acciones que se pueden realizar en un bucket u objeto de S3. Concede estas acciones a las identidades para que puedan actuar en sus recursos de S3. Algunos ejemplos de acciones de S3 son `s3:GetObject` para leer los objetos de un bucket y `s3:PutObject` para escribir objetos en un bucket.

Claves de condición

Además de las acciones, las claves de condición de IAM se limitan a conceder acceso únicamente cuando se cumple una condición. Las claves de condición son opcionales.

> **Note**
>
> En una política de acceso basada en recursos, como una política de bucket, o en una política basada en identidades, puede especificar lo siguiente:

- Una acción o un conjunto de acciones en el elemento `Action` de la instrucción de política.
- En el elemento `Effect` de la instrucción de política, puede especificar `Allow` para conceder las acciones de la lista o puede especificar `Deny` para bloquear las acciones de la lista. Para seguir utilizando la práctica de los privilegios mínimos, las instrucciones `Deny` en el elemento `Effect` de la política de acceso deben ser lo más amplias posibles y las instrucciones `Allow` deben ser lo más reducidas posibles. Los efectos de `Deny` combinados con la acción de `s3:*` son otra buena manera de implementar las prácticas recomendadas de alta de las identidades incluidas en las instrucciones condicionales de la política.
- Una condición clave en el elemento `Condition` de una instrucción de política.

Casos de uso de administración de acceso

Amazon S3 proporciona a los propietarios de los recursos una variedad de herramientas para conceder el acceso. La herramienta de administración de acceso de S3 que utilice depende de los recursos de S3 que desee compartir, las identidades a las que conceda acceso y las acciones que desee permitir o denegar. Puede utilizar una herramienta de administración de acceso de S3 o una combinación de herramientas para administrar el acceso a sus recursos de S3.

En la mayoría de los casos, puede utilizar una política de acceso para administrar los permisos. Una política de acceso puede ser una política basada en recursos que se asocia a un recurso, como un bucket u otro recurso de Amazon S3 (Recursos de S3). Una política de acceso también puede ser una política basada en identidades que se asocia a un usuario, grupo o rol de AWS Identity and Access Management (AIM) en su cuenta. Puede que descubra que una política de bucket funciona mejor para su caso de uso. Para obtener más información, consulte Políticas de buckets para Amazon S3. Como alternativa, con AWS Identity and Access Management (IAM), puede crear usuarios, grupos y roles de IAM dentro de su Cuenta de AWS y administrar su acceso a los buckets y objetos a través de políticas basadas en identidades. Para obtener más información, consulte Políticas basadas en identidad para Amazon S3.

Para ayudarle a explorar estas opciones de administración de acceso, los siguientes son casos de uso comunes de los clientes de Amazon S3 y recomendaciones de cada una de las herramientas de administración de acceso de S3.

El propietario de la Cuenta de AWS quiere compartir buckets únicamente con los usuarios de la misma cuenta

Todas las herramientas de administración de acceso pueden cumplir este caso de uso básico. Recomendamos las siguientes herramientas de administración de acceso para este caso de uso:

- Política de buckets: si quiere conceder acceso a un bucket o a un número reducido de buckets, o bien si sus permisos de acceso a buckets son similares de un bucket a otro, utilice una política de buckets. Con las políticas de buckets, se administra una política para cada bucket. Para obtener más información, consulte Políticas de buckets para Amazon S3.
- Política basada en identidades: si tiene una gran cantidad de buckets con diferentes permisos de acceso para cada bucket y solo tiene que gestionar unos pocos roles de usuario, puede utilizar una política de IAM para los usuarios, grupos o roles. Las políticas de IAM también son una buena opción si gestiona el acceso de los usuarios a otros recursos de AWS, así como a los recursos de Amazon S3. Para obtener más información, consulte Ejemplo 1: propietario del bucket que concede permisos de bucket a sus usuarios.
- Concesiones de acceso a S3: puede utilizar Concesiones de acceso a S3 para conceder acceso a sus buckets, prefijos u objetos de S3. Concesiones de acceso a S3 le permite especificar distintos permisos en el nivel de objetos a escala, mientras que las políticas de bucket tienen un tamaño limitado a 20 KB. Para obtener más información, consulte Introducción a S3 Access Grants.
- Puntos de acceso: puede utilizar puntos de acceso, que son puntos de conexión de red con nombre que están asociados a un bucket. Un bucket puede tener hasta 10 000 puntos de acceso asociados y para cada punto de acceso puede aplicar permisos y controles de red diferenciados para proporcionarle un control detallado sobre el acceso a los objetos de S3. Para obtener más información, consulte Administración del acceso a datos con puntos de acceso de Amazon S3.

El propietario de la Cuenta de AWS quiere compartir buckets u objetos con usuarios de otra cuenta de AWS (entre cuentas)

Para conceder el permiso a otra Cuenta de AWS, debe usar una política de bucket o una de las siguientes herramientas de administración de acceso recomendadas. No puede utilizar una política de acceso basada en identidades para este caso de uso. Para obtener más información sobre la concesión de acceso entre cuentas, consulte ¿Cómo puedo proporcionar acceso entre cuentas a objetos que se encuentran en buckets de Amazon S3?.

Recomendamos las siguientes herramientas de administración de acceso para este caso de uso:

- Política de buckets: con las políticas de buckets, se administra una política para cada bucket. Para obtener más información, consulte Políticas de buckets para Amazon S3.
- Concesiones de acceso a S3: puede utilizar Concesiones de acceso a S3 para conceder permisos entre cuentas a sus buckets, prefijos u objetos de S3. Puede usar Concesiones de acceso a S3 para especificar distintos permisos en el nivel de objetos a escala, mientras que las políticas de buckets tienen un tamaño limitado a 20 KB. Para obtener más información, consulte Introducción a S3 Access Grants.
- Puntos de acceso: puede utilizar puntos de acceso, que son puntos de conexión de red con nombre que están asociados a un bucket. Un bucket puede tener hasta 10 000 puntos de acceso asociados y para cada punto de acceso puede aplicar permisos y controles de red diferenciados para proporcionarle un control detallado sobre el acceso a los objetos de S3. Para obtener más información, consulte Administración del acceso a datos con puntos de acceso de Amazon S3.

El propietario de la Cuenta de AWS o el propietario del bucket debe conceder permisos en el nivel del objeto o del prefijo y estos permisos varían de un objeto a otro o de un prefijo a otro

En una política de buckets, por ejemplo, puede conceder acceso a los objetos de un bucket que comparten un prefijo de nombre clave específico o tienen una etiqueta específica. Puede conceder permisos de lectura sobre objetos que comiencen por el prefijo de nombre de clave `logs/`. Sin embargo, si sus permisos de acceso varían en función del objeto, conceder permisos a objetos individuales con una política de bucket podría no resultar práctico, sobre todo porque las políticas de buckets están limitadas a 20 KB de tamaño.

Recomendamos las siguientes herramientas de administración de acceso para este caso de uso:

- Concesiones de acceso a S3: puede usar Concesiones de acceso a S3 para administrar los permisos en el nivel de objetos o de prefijos. A diferencia de las políticas de buckets, puede usar Concesiones de acceso a S3 para especificar distintos permisos en el nivel de objetos a escala. Las políticas de bucket tienen un límite de tamaño de 20 KB. Para obtener más información, consulte Introducción a S3 Access Grants.
- Puntos de acceso: puede usar puntos de acceso para administrar permisos en el nivel de objetos o de prefijos. Los puntos de acceso son puntos de conexión de red con nombre que están asociados a los buckets. Un bucket puede tener hasta 10 000 puntos de acceso asociados y para cada punto de acceso puede aplicar permisos y controles de red diferenciados para proporcionarle un control detallado sobre el acceso a los objetos de S3. Para obtener más información, consulte Administración del acceso a datos con puntos de acceso de Amazon S3.

- ACL: no recomendamos el uso de listas de control de acceso (ACL), sobre todo porque las ACL están limitadas a 100 concesiones por objeto. Sin embargo, si opta por activar las ACL, en los ajustes del buckets, establezca la propiedad del objeto como Propietario del bucket preferido y ACL habilitadas. Con esta configuración, nuevos objetos que se escriben con la ACL predefinida `bucket-owner-full-control` pertenecen automáticamente al propietario del bucket en lugar del escritor del objeto. A continuación, puede usar ACL de objetos, que son una política de acceso con formato XML, para conceder a otros usuarios acceso al objeto. Para obtener más información, consulte Información general de las Listas de control de acceso (ACL).

El propietario de la Cuenta de AWS o propietario del bucket quiere limitar el acceso al bucket únicamente a ID de cuenta específicos

Recomendamos las siguientes herramientas de administración de acceso para este caso de uso:

- Política de buckets: con las políticas de buckets, se administra una política para cada bucket. Para obtener más información, consulte Políticas de buckets para Amazon S3.
- Puntos de acceso: los puntos de acceso son puntos de conexión de red con nombre que están asociados a un bucket. Un bucket puede tener hasta 10 000 puntos de acceso asociados y para cada punto de acceso puede aplicar permisos y controles de red diferenciados para proporcionarle un control detallado sobre el acceso a los objetos de S3. Para obtener más información, consulte Administración del acceso a datos con puntos de acceso de Amazon S3.

El propietario de la Cuenta de AWS o el propietario del bucket quiere puntos de conexión distintos para cada usuario o aplicación que acceda a sus datos

Recomendamos las siguientes herramientas de administración de acceso para este caso de uso:

- Puntos de acceso: los puntos de acceso son puntos de conexión de red con nombre que están asociados a un bucket. Un bucket puede tener hasta 10 000 puntos de acceso asociados y para cada punto de acceso puede aplicar permisos y controles de red diferenciados para proporcionarle un control detallado sobre el acceso a los objetos de S3. Cada punto de acceso aplica una política de punto de acceso personalizada que funciona en conjunción con la política de bucket asociada al bucket subyacente. Para obtener más información, consulte Administración del acceso a datos con puntos de acceso de Amazon S3.

El propietario de la Cuenta de AWS o el propietario del bucket debe administrar el acceso desde los puntos de conexión de nube privada virtual (VPC) para S3

Los punto de conexión de nube privada virtual (VPC) de Amazon S3 son entidades lógicas dentro de una VPC que permiten la conectividad solo a Amazon S3. Recomendamos las siguientes herramientas de administración de acceso para este caso de uso:

- Buckets en una configuración de VPC: puede usar una política de bucket para controlar quién puede acceder a los buckets y a qué puntos de conexión de VPC pueden acceder. Para obtener más información, consulte Control del acceso desde puntos de enlace de la VPC con políticas de bucket.
- Puntos de acceso: si opta por configurar puntos de acceso, puede usar una política de puntos de acceso. Puede configurar cualquier punto de acceso para que acepte solo las solicitudes procedentes de una nube privada virtual (VPC) con el fin de restringir el acceso a los datos de Amazon S3 a una red privada. También puede configurar los parámetros de bloqueo de acceso público para cada punto de acceso. Para obtener más información, consulte Administración del acceso a datos con puntos de acceso de Amazon S3.

El propietario de la Cuenta de AWS o el propietario del bucket debe hacer que un sitio web estático esté disponible públicamente

Con S3, puede alojar un sitio web estático y permitir que cualquier persona vea el contenido del sitio web, que se aloja en un bucket de S3.

Recomendamos las siguientes herramientas de administración de acceso para este caso de uso:

- Amazon CloudFront: esta solución le permite alojar un sitio web estático de Amazon S3 para el público y, al mismo tiempo, seguir bloqueando todo el acceso público al contenido de un bucket. Si desea mantener los cuatro ajustes de bloqueo de acceso público de S3 activados y alojar un sitio web estático de S3, puede utilizar el control de acceso de origen (OAC) de Amazon CloudFront. Amazon CloudFront proporciona las capacidades necesarias para configurar un sitio web estático seguro. Además, los sitios web estáticos de Amazon S3 que no utilizan esta solución solo admiten puntos de conexión HTTP. CloudFront utiliza el almacenamiento duradero de Amazon S3 a la vez que proporciona encabezados de seguridad adicionales, tales como HTTPS. HTTPS agrega seguridad al cifrar una solicitud HTTP normal y proteger contra ataques cibernéticos comunes.

  Para obtener información, consulte Introducción a un sitio web estático seguro en la guía para desarrolladores de Amazon CloudFront.

- Hacer que su bucket de Amazon S3 sea de acceso público: puede configurar un bucket para que se utilice como un sitio web estático de acceso público.

  > **Warning**
  >
  > No recomendamos este método. En su lugar, le recomendamos que utilice sitios web estáticos de Amazon S3 como parte de Amazon CloudFront. Para obtener más información, consulte la opción anterior, o bien consulte Introducción a un sitio web seguro estático.

  Para crear un sitio web estático de Amazon S3 sin Amazon CloudFront, primero debe desactivar todos los ajustes de bloqueo de acceso público. Al escribir la política del bucket para el sitio web estático, asegúrese de permitir solo acciones de `s3:GetObject` y no permisos de `ListObject` o `PutObject`. Esto ayuda a garantizar que los usuarios no puedan ver todos los objetos del bucket ni agregar su propio contenido. Para obtener más información, consulte Configurar permisos para el acceso a sitios web.

El propietario de la Cuenta de AWS o el propietario del bucket quiere que el contenido de un bucket esté disponible públicamente

Al crear un nuevo bucket de Amazon S3, el ajuste Bloqueo de acceso público está habilitado de forma predeterminada. Para obtener más información acerca del bloqueo del acceso público, consulte Bloquear el acceso público a su almacenamiento de Amazon S3.

No recomendamos que permita el acceso público a su bucket. Sin embargo, si debe hacerlo para un caso de uso concreto, le recomendamos la siguiente herramienta de administración de acceso:

- Disable Block Public Access setting: el propietario de un bucket puede permitir que sin autenticar al bucket. Por ejemplo, se permiten solicitudes PUT Object no autenticadas cuando un bucket tiene una política de bucket pública o cuando una ACL de bucket concede acceso público. Todas las solicitudes no autenticadas las realizan otros usuarios de AWSarbitrarios o incluso usuarios anónimos no autenticados. En las ACL, este usuario se representa mediante el ID de usuario canónico específico`65a011a29cdf8ec533ec3d1ccaae921c`. Si un objeto se carga en `WRITE` o `FULL_CONTROL`, se concede acceso específicamente al grupo Todos los usuarios o al usuario anónimo. Para obtener más información acerca de las políticas de bucket públicas y las listas de control de acceso (ACL) públicas, consulte Qué significa "pública".

El propietario de la Cuenta de AWS o el propietario del bucket ha superado los límites de tamaño de la política de acceso

Tanto las políticas de buckets como las políticas basadas en identidad tienen un límite de tamaño de 20 KB. Si sus requisitos de permiso de acceso son complejos, es posible que supere este límite de tamaño.

Recomendamos las siguientes herramientas de administración de acceso para este caso de uso:

- Puntos de acceso: utilice puntos de acceso si esto funciona con su caso de uso. Con puntos de acceso, cada bucket tiene varios puntos de conexión de red con nombre, cada uno con su propia política de punto de acceso que funciona con la política de bucket subyacente. Sin embargo, los puntos de acceso solo pueden actuar en objetos, no en buckets, y no admiten la replicación entre regiones. Para obtener más información, consulte Administración del acceso a datos con puntos de acceso de Amazon S3.
- Concesiones de acceso a S3: utilice Concesiones de acceso a S3, que admite una gran cantidad de concesiones que dan acceso a buckets, prefijos u objetos. Para obtener más información, consulte Introducción a S3 Access Grants.

El rol del propietario de la Cuenta de AWS o de administrador quiere conceder acceso a un bucket, prefijo u objeto directamente a los usuarios o grupos de un directorio corporativo

En lugar de administrar los usuarios, los grupos y los roles mediante AWS Identity and Access Management (IAM), puede añadir el directorio corporativo a AWS IAM Identity Center. Para obtener más información consulte What Is IAM Identity Center?.

Después de agregar el directorio corporativo a AWS IAM Identity Center, le recomendamos que utilice la siguiente herramienta de administración de acceso para conceder acceso a sus recursos de S3 a identidades del directorio corporativo:

- Concesiones de acceso a S3: utilice Concesiones de acceso a S3, que admite la concesión de acceso a usuarios o roles de su directorio corporativo. Para obtener más información, consulte Introducción a S3 Access Grants.

El propietario de la Cuenta de AWS o el propietario del bucket quiere conceder acceso al servicio AWS CloudFront para escribir registros de CloudFront en un bucket de S3

Recomendamos la siguiente herramienta de administración de acceso para este caso de uso:

- ACL del bucket: el único caso de uso recomendado para las ACL de bucket es para conceder permisos a ciertos Servicios de AWS, como la cuenta `awslogsdelivery` de Amazon CloudFront. Al crear o actualizar una distribución y habilitar el registro de CloudFront, CloudFront actualiza la ACL del bucket para conceder permisos `FULL_CONTROL` a la cuenta `awslogsdelivery` para que escriba registros en el bucket. Para obtener más información, consulte Permisos necesarios para configurar el registro estándar y acceder a los archivos de registro en la Guía para desarrolladores de Amazon CloudFront. Si el bucket que almacena los registros utiliza el ajuste Aplicada al propietario del bucket para Propiedad de objetos de S3 para desactivar las ACL, CloudFront no puede escribir registros en el bucket. Para obtener más información, consulte Control de la propiedad de los objetos y desactivación de las ACL del bucket.

Como propietario del bucket, quiere mantener el control total de los objetos que otros usuarios añaden al bucket

Puede conceder acceso a otras cuentas para que carguen objetos en el bucket mediante una política de bucket, un punto de acceso o Concesiones de acceso a S3 Si ha concedido acceso entre cuentas a su bucket, puede asegurarse de que todos los objetos que se suban al bucket permanezcan bajo su control total.

Recomendamos la siguiente herramienta de administración de acceso para este caso de uso:

- Propiedad del objeto: mantenga el ajuste Propiedad del objeto en el nivel de buckets en el ajuste predeterminado Aplicada al propietario del bucket.

Solución de problemas de administración de accesos

Los siguientes recursos pueden ayudarle a solucionar problemas con la administración de accesos de S3:

Solución de errores de acceso denegado (403 Prohibido)

Si tiene problemas de denegación de acceso, compruebe los ajustes en la cuenta y el bucket. Además, compruebe la característica de administración de acceso que está utilizando para conceder accesos y asegurarse de que la política, el ajuste o la configuración son correctos. Para obtener más información sobre las causas comunes de los errores de acceso denegado (403 Prohibido) en Amazon S3, consulte Solucionar errores de acceso denegado (403 Prohibido) en Amazon S3.

Analizador de acceso de IAM para S3

Si no quiere que ninguno de sus recursos esté disponible públicamente o si quiere limitar el acceso público a sus recursos, puede utilizar Analizador de acceso de IAM para S3. En la consola de Amazon S3, puede utilizar el Analizador de acceso de IAM para S3 para revisar todos los buckets que tienen listas de control de acceso (ACL) de buckets, políticas de buckets o políticas de puntos de acceso que otorguen acceso público o compartido. Analizador de acceso de IAM para S3 le avisa de los buckets que están configurados para permitir el acceso a cualquier usuario de Internet u otras Cuentas de AWS, incluidas aquellas Cuentas de AWS ajenas a la organización. Para cada bucket público o compartido, recibirá resultados que le informarán del origen y el nivel de acceso público o compartido.

En Analizador de acceso de IAM para S3, puede bloquear todo el acceso público a un bucket con una sola acción. Le recomendamos que bloquee todo el acceso público a sus buckets a menos que necesite acceso público para admitir un caso de uso específico. Antes de bloquear todo el acceso público, asegúrese de que las aplicaciones seguirán funcionando correctamente sin ese acceso público. Para obtener más información, consulte Bloquear el acceso público a su almacenamiento de Amazon S3.

También puede revisar a fondo los ajustes de permisos de buckets para configurar niveles detallados de acceso. Para casos de uso específicos y verificados que requieren acceso público o compartido, puede reconocer y registrar su intención de que el bucket continúe siendo público o compartido archivando los resultados del bucket. Puede volver a visitar y modificar estas configuraciones de bucket en cualquier momento. También puede descargar sus resultados en un informe CSV con fines de auditoría.

Analizador de acceso de IAM para S3 está disponible sin coste adicional en la consola de Amazon S3. Analizador de acceso de IAM para S3 cuenta con la tecnología de Analizador de acceso de AWS Identity and Access Management (IAM). Para utilizar el Analizador de acceso de IAM para S3 en la consola de Amazon S3, debe ir a la consola de IAM y crear un analizador de cuenta en Analizador de acceso de IAM por cada región.

Para obtener más información acerca de Analizador de acceso de IAM para S3, consulte Revisión del acceso al bucket mediante Analizador de acceso de IAM para S3.

Registro y monitorización

La monitorización es una parte importante del mantenimiento de la fiabilidad, la disponibilidad y el rendimiento de las soluciones de Amazon S3 para que pueda depurar con más facilidad un fallo de acceso. El registro puede proporcionar información sobre los errores que los usuarios reciben, qué

solicitudes se realizan y en qué momento. AWS proporciona varias herramientas para monitorear los recursos de Amazon S3, como las siguientes:

- AWS CloudTrail
- Registros de acceso de Amazon S3
- AWS Trusted Advisor
- Amazon CloudWatch

Para obtener más información, consulte Registro y monitoreo en Amazon S3.

Temas

Administración de identidades y accesos para Amazon S3

AWS Identity and Access Management (IAM) es un Servicio de AWS que ayuda a los administradores a controlar de forma segura el acceso a los recursos de AWS. Los administradores de IAM controlan a qué personas se puede autenticar (pueden iniciar sesión) y autorizar (tienen permisos) para utilizar recursos de Amazon S3. IAM es un Servicio de AWS que se puede utilizar sin cargo adicional.

> **Note**
>
> Para obtener más información sobre el uso de la clase de almacenamiento Amazon S3 Express One Zone con buckets de directorio, consulte ¿Qué es S3 Express One Zone? y Buckets de directorio.

Temas

Público

La forma en que utilice AWS Identity and Access Management (IAM) varía en función del trabajo que realice en Amazon S3.

Usuario de servicio: si utiliza el servicio de Amazon S3 para realizar su trabajo, el administrador le proporciona las credenciales y los permisos que necesita. A medida que utilice más características de Amazon S3 para realizar el trabajo, es posible que necesite permisos adicionales. Entender cómo se administra el acceso puede ayudarlo a solicitar los permisos correctos al administrador. Si no puede acceder a una característica de Amazon S3, consulte Solución de problemas de identidad y acceso de Amazon S3.

Administrador de servicio: si está a cargo de los recursos de Amazon S3 de la empresa, probablemente tenga acceso completo a Amazon S3. Su trabajo consiste en determinar a qué características y recursos de Amazon S3 deben acceder los usuarios del servicio. Luego, debe enviar solicitudes a su administrador de IAM para cambiar los permisos de los usuarios de su servicio. Revise la información de esta página para conocer los conceptos básicos de IAM. Para obtener más información acerca de cómo la empresa puede utilizar IAM con Amazon S3, consulte Cómo funciona Amazon S3 con IAM.

Administrador de IAM: si es un administrador de IAM, es posible que desee obtener información acerca de cómo escribir políticas para administrar el acceso a Amazon S3. Para ver ejemplos de

políticas basadas en identidad de Amazon S3 que puede utilizar en IAM, consulte Políticas basadas en identidad para Amazon S3.

Autenticación con identidades

La autenticación es la manera de iniciar sesión en AWS mediante credenciales de identidad. Debe estar autenticado (haber iniciado sesión en AWS) como Usuario raíz de la cuenta de AWS, como un usuario de IAM o asumiendo un rol de IAM.

Puede iniciar sesión en AWS como una identidad federada mediante las credenciales proporcionadas a través de una fuente de identidad. AWS IAM Identity Center Los usuarios (del Centro de identidades de IAM), la autenticación de inicio de sesión único de su empresa y sus credenciales de Google o Facebook son ejemplos de identidades federadas. Al iniciar sesión como una identidad federada, su administrador habrá configurado previamente la federación de identidades mediante roles de IAM. Cuando accede a AWS mediante la federación, está asumiendo un rol de forma indirecta.

Según el tipo de usuario que sea, puede iniciar sesión en AWS Management Console o en el portal de acceso AWS. Para obtener más información sobre el inicio de sesión en AWS, consulte Cómo iniciar sesión en su Cuenta de AWS en la Guía del usuario de AWS Sign-In.

Si accede a AWS mediante programación, AWS proporciona un kit de desarrollo de software (SDK) y una interfaz de la línea de comandos (CLI) para firmar criptográficamente las solicitudes mediante el uso de las credenciales. Si no usa las herramientas de AWS, debe firmar las solicitudes. Para obtener más información sobre la firma de solicitudes, consulte Firma de solicitudes API de AWS en la Guía del usuario de IAM.

Independientemente del método de autenticación que use, es posible que deba proporcionar información de seguridad adicional. Por ejemplo, AWS le recomienda el uso de la autenticación multifactor (MFA) para aumentar la seguridad de su cuenta. Para obtener más información, consulte Autenticación multifactor en la Guía del usuario de AWS IAM Identity Center y Uso de la autenticación multifactor (MFA) en AWSen la Guía del usuario de IAM.

Usuario raíz de Cuenta de AWS

Cuando se crea una Cuenta de AWS, se comienza con una identidad de inicio de sesión que tiene acceso completo a todos los recursos y Servicios de AWS de la cuenta. Esta identidad recibe el nombre de usuario raíz de la Cuenta de AWS y se accede a ella iniciando sesión con el email y la contraseña que utilizó para crear la cuenta. Recomendamos encarecidamente que no utilice el usuario raíz para sus tareas diarias. Proteja las credenciales del usuario raíz y utilícelas solo para las

tareas que solo el usuario raíz pueda realizar. Para ver la lista completa de las tareas que requieren que inicie sesión como usuario raíz, consulte Tareas que requieren credenciales de usuario raíz en la Guía del usuario de IAM.

Identidad federada

Como práctica recomendada, solicite que los usuarios humanos, incluidos los que requieren acceso de administrador, utilicen la federación con un proveedor de identidades para acceder a los Servicios de AWS utilizando credenciales temporales.

Una identidad federada es un usuario del directorio de usuarios de su empresa, un proveedor de identidad web, el AWS Directory Service, el directorio del Identity Center, o cualquier usuario que acceda a Servicios de AWS utilizando credenciales proporcionadas a través de una fuente de identidad. Cuando identidades federadas acceden a Cuentas de AWS, asumen roles y los roles proporcionan credenciales temporales.

Para una administración de acceso centralizada, le recomendamos que utilice AWS IAM Identity Center. Puede crear usuarios y grupos en el IAM Identity Center o puede conectarse y sincronizar con un conjunto de usuarios y grupos de su propia fuente de identidad para usarlos en todas sus aplicaciones y Cuentas de AWS. Para obtener más información, consulte ¿Qué es el Centro de identidades de IAM? en la Guía del usuario de AWS IAM Identity Center.

Usuarios y grupos de IAM

Un usuario de IAM es una identidad de la Cuenta de AWSque dispone de permisos específicos para una sola persona o aplicación. Siempre que sea posible, recomendamos emplear credenciales temporales, en lugar de crear usuarios de IAM que tengan credenciales de larga duración como contraseñas y claves de acceso. No obstante, si tiene casos de uso específicos que requieran credenciales de larga duración con usuarios de IAM, recomendamos rotar las claves de acceso. Para más información, consulte Rotar las claves de acceso periódicamente para casos de uso que requieran credenciales de larga duración en la Guía del usuario de IAM.

Un grupo de IAM es una identidad que especifica un conjunto de usuarios de IAM. No puede iniciar sesión como grupo. Puede usar los grupos para especificar permisos para varios usuarios a la vez. Los grupos facilitan la administración de los permisos de grandes conjuntos de usuarios. Por ejemplo, podría tener un grupo cuyo nombre fuese IAMAdmins y conceder permisos a dicho grupo para administrar los recursos de IAM.

Los usuarios son diferentes de los roles. Un usuario se asocia exclusivamente a una persona o aplicación, pero la intención es que cualquier usuario pueda asumir un rol que necesite. Los usuarios

tienen credenciales permanentes a largo plazo y los roles proporcionan credenciales temporales. Para más información, consulte Cuándo crear un usuario de IAM (en lugar de un rol) en la Guía del usuario de IAM.

Roles de IAM

Un rol de IAM es una identidad de la Cuenta de AWSque dispone de permisos específicos. Es similar a un usuario de IAM, pero no está asociado a una determinada persona. Puede asumir temporalmente un rol de IAM en la AWS Management Console cambiando de roles. Puede asumir un rol llamando a una operación de la AWS CLI o de la API de AWS, o utilizando una URL personalizada. Para más información sobre los métodos para el uso de roles, consulte Uso de roles de IAM en la Guía del usuario de IAM.

Los roles de IAM con credenciales temporales son útiles en las siguientes situaciones:

- Acceso de usuario federado: para asignar permisos a una identidad federada, puede crear un rol y definir sus permisos. Cuando se autentica una identidad federada, se asocia la identidad al rol y se le conceden los permisos define el rol. Para obtener información acerca de roles para federación, consulte Creación de un rol para un proveedor de identidades de terceros en la Guía del usuario de IAM. Si utiliza IAM Identity Center, debe configurar un conjunto de permisos. IAM Identity Center correlaciona el conjunto de permisos con un rol en IAM para controlar a qué pueden acceder las identidades después de autenticarse. Para obtener información acerca de los conjuntos de permisos, consulte Conjuntos de permisos en la Guía del usuario de AWS IAM Identity Center.
- Permisos de usuario de IAM temporales: un usuario de IAM puede asumir un rol de IAM para recibir temporalmente permisos distintos que le permitan realizar una tarea concreta.
- Acceso entre cuentas: puede utilizar un rol de IAM para permitir que alguien (una entidad principal de confianza) de otra cuenta acceda a los recursos de la cuenta. Los roles son la forma principal de conceder acceso entre cuentas. No obstante, con algunos Servicios de AWS se puede adjuntar una política directamente a un recurso (en lugar de utilizar un rol como representante). Para obtener información acerca de la diferencia entre los roles y las políticas basadas en recursos para el acceso entre cuentas, consulte Acceso a recursos entre cuentas en IAM en la Guía del usuario de IAM.
- Acceso entre servicios: algunos Servicios de AWS utilizan características de otros Servicios de AWS. Por ejemplo, cuando realiza una llamada en un servicio, es común que ese servicio ejecute aplicaciones en Amazon EC2 o almacene objetos en Amazon S3. Es posible que un servicio haga esto usando los permisos de la entidad principal, usando un rol de servicio o usando un rol vinculado a servicios.

  - Reenviar sesiones de acceso (FAS): cuando utiliza un rol o un usuario de IAM para llevar a cabo acciones en AWS, se le considera una entidad principal. Cuando utiliza algunos servicios, es posible que realice una acción que desencadene otra acción en un servicio diferente. FAS utiliza los permisos de la entidad principal para llamar a un Servicio de AWS, combinados con el Servicio de AWS solicitante para realizar solicitudes a servicios posteriores. Las solicitudes de FAS solo se realizan cuando un servicio recibe una solicitud que requiere interacciones con otros Servicios de AWS o recursos para completarse. En este caso, debe tener permisos para realizar ambas acciones. Para obtener información sobre las políticas a la hora de realizar solicitudes de FAS, consulte Reenviar sesiones de acceso.
  - Rol de servicio: un rol de servicio es un rol de IAM que adopta un servicio para realizar acciones en su nombre. Un administrador de IAM puede crear, modificar y eliminar un rol de servicio desde IAM. Para obtener más información, consulte Creación de un rol para delegar permisos a un Servicio de AWS en la Guía del usuario de IAM.
  - Rol vinculado a los servicios: un rol vinculado a servicios es un tipo de rol de servicio que está vinculado a un Servicio de AWS. El servicio puede asumir el rol para realizar una acción en su nombre. Los roles vinculados a servicios aparecen en la Cuenta de AWS y son propiedad del servicio. Un administrador de IAM puede ver, pero no editar, los permisos de los roles vinculados a servicios.
- Aplicaciones que se ejecutan en Amazon EC2: puede utilizar un rol de IAM que le permita administrar credenciales temporales para las aplicaciones que se ejecutan en una instancia de EC2 y realizan solicitudes a la AWS CLI o a la API de AWS. Es preferible hacerlo de este modo a almacenar claves de acceso en la instancia de EC2. Para asignar un rol de AWS a una instancia de EC2 y ponerla a disposición de todas las aplicaciones, cree un perfil de instancia adjuntado a la instancia. Un perfil de instancia contiene el rol y permite a los programas que se ejecutan en la instancia de EC2 obtener credenciales temporales. Para más información, consulte Uso de un rol de IAM para conceder permisos a aplicaciones que se ejecutan en instancias Amazon EC2 en la Guía del usuario de IAM.

Para obtener información sobre el uso de los roles de IAM, consulte Cuándo crear un rol de IAM (en lugar de un usuario) en la Guía del usuario de IAM.

Administración de acceso mediante políticas

Para controlar el acceso en AWS, se crean políticas y se adjuntan a identidades o recursos de AWS. Una política es un objeto de AWS que, cuando se asocia a una identidad o un recurso, define sus permisos. AWS evalúa estas políticas cuando una entidad principal (sesión de rol, usuario o usuario

raíz) realiza una solicitud. Los permisos en las políticas determinan si la solicitud se permite o se deniega. La mayoría de las políticas se almacenan en AWS como documentos JSON. Para obtener más información sobre la estructura y el contenido de los documentos de política JSON, consulte Información general de políticas JSON en la Guía del usuario de IAM.

Los administradores pueden utilizar las políticas JSON de AWS para especificar quién tiene acceso a qué. Es decir, qué entidad principal puede realizar acciones en qué recursos y en qué condiciones.

De forma predeterminada, los usuarios y los roles no tienen permisos. Un administrador de IAM puede crear políticas de IAM para conceder permisos a los usuarios para realizar acciones en los recursos que necesitan. A continuación, el administrador puede añadir las políticas de IAM a roles y los usuarios pueden asumirlos.

Las políticas de IAM definen permisos para una acción independientemente del método que se utilice para realizar la operación. Por ejemplo, suponga que dispone de una política que permite la acción `iam:GetRole`. Un usuario con dicha política puede obtener información del usuario de la AWS Management Console, la AWS CLI o la API de AWS.

Políticas basadas en identidades

Las políticas basadas en identidad son documentos de políticas de permisos JSON que puede asociar a una identidad, como un usuario de IAM, un grupo de usuarios o un rol. Estas políticas controlan qué acciones pueden realizar los usuarios y los roles, en qué recursos y en qué condiciones. Para obtener más información sobre cómo crear una política basada en identidad, consulte Creación de políticas de IAM en la Guía del usuario de IAM.

Las políticas basadas en identidades pueden clasificarse además como políticas insertadas o políticas administradas. Las políticas insertadas se integran directamente en un único usuario, grupo o rol. Las políticas administradas son políticas independientes que puede adjuntar a varios usuarios, grupos y roles de su Cuenta de AWS. Las políticas administradas incluyen las políticas administradas de AWS y las políticas administradas por el cliente. Para más información sobre cómo elegir una política administrada o una política insertada, consulte Elegir entre políticas administradas y políticas insertadas en la Guía del usuario de IAM.

Políticas basadas en recursos

Las políticas basadas en recursos son documentos de política JSON que se asocian a un recurso. Ejemplos de políticas basadas en recursos son las políticas de confianza de roles de IAM y las políticas de bucket de Amazon S3. En los servicios que admiten políticas basadas en recursos, los

administradores de servicios pueden utilizarlos para controlar el acceso a un recurso específico. Para el recurso al que se asocia la política, la política define qué acciones puede realizar una entidad principal especificada en ese recurso y en qué condiciones. Debe especificar una entidad principal en una política en función de recursos. Las entidades principales pueden incluir cuentas, usuarios, roles, usuarios federados o Servicios de AWS.

Las políticas basadas en recursos son políticas insertadas que se encuentran en ese servicio. No se puede utilizar políticas de IAM administradas de AWS en una política basada en recursos.

Listas de control de acceso (ACL)

Las listas de control de acceso (ACL) controlan qué entidades principales (miembros de cuentas, usuarios o roles) tienen permisos para acceder a un recurso. Las ACL son similares a las políticas basadas en recursos, aunque no utilizan el formato de documento de políticas JSON.

Amazon S3, AWS WAF y VPC son ejemplos de servicios que admiten las ACL. Para obtener más información sobre las ACL, consulte Información general de Lista de control de acceso (ACL) en la Guía para desarrolladores de Amazon Simple Storage Service.

Otros tipos de políticas

AWS admite otros tipos de políticas adicionales menos frecuentes. Estos tipos de políticas pueden establecer el máximo de permisos que los tipos de políticas más frecuentes le conceden.

- Límites de permisos: un límite de permisos es una característica avanzada que le permite establecer los permisos máximos que una política basada en identidad puede conceder a una entidad de IAM (usuario o rol de IAM). Puede establecer un límite de permisos para una entidad. Los permisos resultantes son la intersección de las políticas basadas en la identidad de la entidad y los límites de permisos. Las políticas basadas en recursos que especifiquen el usuario o rol en el campo `Principal` no estarán restringidas por el límite de permisos. Una denegación explícita en cualquiera de estas políticas anulará el permiso. Para obtener más información sobre los límites de los permisos, consulte Límites de permisos para las entidades de IAM en la Guía del usuario de IAM.
- Políticas de control de servicio (SCP): las SCP son políticas de JSON que especifican los permisos máximos de una organización o una unidad organizativa en AWS Organizations. AWS Organizations es un servicio que le permite agrupar y administrar de manera centralizada varias Cuentas de AWS que posea su empresa. Si habilita todas las características en una empresa, entonces podrá aplicar políticas de control de servicio (SCP) a una o todas sus cuentas. Una SCP limita los permisos para las entidades de las cuentas de miembros, incluido cada Usuario raíz de

la cuenta de AWS. Para obtener más información acerca de Organizations y las SCP, consulte Funcionamiento de las SCP en la Guía del usuario de AWS Organizations.

- Políticas de sesión: las políticas de sesión son políticas avanzadas que se pasan como parámetro cuando se crea una sesión temporal mediante programación para un rol o un usuario federado. Los permisos de la sesión resultantes son la intersección de las políticas basadas en identidades del rol y las políticas de la sesión. Los permisos también pueden proceder de una política en función de recursos. Una denegación explícita en cualquiera de estas políticas anulará el permiso. Para más información, consulte Políticas de sesión en la Guía del usuario de IAM.

Varios tipos de políticas

Cuando se aplican varios tipos de políticas a una solicitud, los permisos resultantes son más complicados de entender. Para obtener información acerca de cómo AWS decide si permitir o no una solicitud cuando hay varios tipos de políticas implicados, consulte Lógica de evaluación de políticas en la Guía del usuario de IAM.

Cómo funciona Amazon S3 con IAM

Antes de utilizar IAM para administrar el acceso a Amazon S3, obtenga información sobre qué características de IAM se encuentran disponibles para su uso con Amazon S3.

Características de IAM que puede utilizar con Amazon S3

Característica de IAM	Compatibilidad con Amazon S3
Políticas basadas en identidades	Sí
Políticas basadas en recursos	Sí
Acciones de políticas	Sí
Recursos de políticas	Sí
Claves de condición de política (específicas del servicio)	Sí
ACL	Sí
ABAC (etiquetas en políticas)	Parcial

Característica de IAM	Compatibilidad con Amazon S3
Credenciales temporales	Sí
Sesiones de acceso directo (FAS)	Sí
Roles de servicio	Sí
Roles vinculados al servicio	Parcial

Para obtener una perspectiva general sobre cómo funcionan Amazon S3 y otros servicios de AWS con la mayoría de las características de IAM, consulte Servicios de AWS que funcionan con IAM en la Guía del usuario de IAM.

Políticas basadas en identidad para Amazon S3

Compatibilidad con las políticas basadas en identidad: sí

Las políticas basadas en identidad son documentos de políticas de permisos JSON que puede asociar a una identidad, como un usuario de IAM, un grupo de usuarios o un rol. Estas políticas controlan qué acciones pueden realizar los usuarios y los roles, en qué recursos y en qué condiciones. Para obtener más información sobre cómo crear una política basada en identidad, consulte Creación de políticas de IAM en la Guía del usuario de IAM.

Con las políticas basadas en identidades de IAM, puede especificar las acciones y los recursos permitidos o denegados, así como las condiciones en las que se permiten o deniegan las acciones. No es posible especificar la entidad principal en una política basada en identidad porque se aplica al usuario o rol al que está adjunto. Para más información sobre los elementos que puede utilizar en una política de JSON, consulte Referencia de los elementos de las políticas de JSON de IAM en la Guía del usuario de IAM.

Ejemplos de políticas basadas en identidad para Amazon S3

Para ver ejemplos de políticas basadas en identidad de Amazon S3, consulte Políticas basadas en identidad para Amazon S3.

Políticas basadas en recursos dentro de Amazon S3

Compatibilidad con las políticas basadas en recursos: sí

Las políticas basadas en recursos son documentos de política JSON que se asocian a un recurso. Ejemplos de políticas basadas en recursos son las políticas de confianza de roles de IAM y las políticas de bucket de Amazon S3. En los servicios que admiten políticas basadas en recursos, los administradores de servicios pueden utilizarlos para controlar el acceso a un recurso específico. Para el recurso al que se asocia la política, la política define qué acciones puede realizar una entidad principal especificada en ese recurso y en qué condiciones. Debe especificar una entidad principal en una política en función de recursos. Las entidades principales pueden incluir cuentas, usuarios, roles, usuarios federados o Servicios de AWS.

Para habilitar el acceso entre cuentas, puede especificar toda una cuenta o entidades de IAM de otra cuenta como la entidad principal de una política en función de recursos. Añadir a una política en función de recursos una entidad principal entre cuentas es solo una parte del establecimiento de una relación de confianza. Cuando la entidad principal y el recurso se encuentran en Cuentas de AWS diferentes, un administrador de IAM de la cuenta de confianza también debe conceder a la entidad principal (usuario o rol) permiso para acceder al recurso. Para conceder el permiso, adjunte la entidad a una política basada en identidad. Sin embargo, si la política en función de recursos concede el acceso a una entidad principal de la misma cuenta, no es necesaria una política basada en identidad adicional. Para más información, consulte Cross account resource access in IAM en la Guía del usuario de IAM.

El servicio de Amazon S3 admite políticas de bucket, políticas de puntos de acceso y concesiones de acceso:

- Las políticas de bucket son políticas basadas en recursos adjuntas a un bucket de Amazon S3. Una política de bucket define qué entidades principales pueden realizar acciones en el bucket.
- Las políticas de puntos de acceso son políticas basadas en recursos que se evalúan junto con la política de bucket subyacente.
- Las concesiones de acceso son un modelo simplificado para definir los permisos de acceso a los datos en Amazon S3 por prefijo, bucket u objeto. Para obtener información sobre las concesiones de acceso de S3, consulte Administración del acceso con S3 Access Grants.

Entidades principales de las políticas de bucket

El elemento `Principal` especifica el usuario, la cuenta, el servicio u otra entidad con acceso permitido o denegado para un recurso. A continuación se proporcionan ejemplos de la especificación del elemento `Principal`. Para obtener más información, consulte Principal en la guía del usuario de IAM.

Concesión de permisos a una Cuenta de AWS

Para conceder permisos a una Cuenta de AWS, use el siguiente formato para identificar la cuenta.

```
"AWS":"account-ARN"
```

A continuación se muestran algunos ejemplos.

```
"Principal":{"AWS":"arn:aws:iam::AccountIDWithoutHyphens:root"}
```

```
"Principal":{"AWS":
["arn:aws:iam::AccountID1WithoutHyphens:root","arn:aws:iam::AccountID2WithoutHyphens:root"]}
```

Conceder permisos a un usuario de IAM

Para conceder permisos a un usuario de IAM dentro de la cuenta, debe proporcionar el par nombre-valor `"AWS":"user-ARN"`.

```
"Principal":{"AWS":"arn:aws:iam::account-number-without-hyphens:user/username"}
```

Para obtener ejemplos detallados que proporcionan instrucciones paso a paso, consulte Ejemplo 1: propietario del bucket que concede permisos de bucket a sus usuarios y Ejemplo 3: propietario del bucket que concede a sus usuarios permisos para objetos que no posee.

> **Note**
>
> Si se elimina una identidad de IAM después de actualizar la política de bucket, la política de bucket mostrará un identificador único en el elemento de la entidad principal en lugar de un ARN. Estos ID únicos nunca se reutilizan, por lo que puede eliminar de forma segura las entidades principales con identificadores únicos de todas las instrucciones de política. Para obtener más información acerca de los identificadores únicos, consulte Identificadores de IAM en la Guía del usuario de IAM.

Conceder permisos anónimos

> **Warning**
>
> Extreme las precauciones a la hora de otorgar acceso anónimo a su bucket de Amazon S3. Al otorgar acceso anónimo, cualquier persona puede acceder a su bucket. Se recomienda encarecidamente que no otorgue nunca ningún tipo de acceso de escritura anónimo en su bucket de S3.

Para conceder permisos a todos los usuarios, lo que también se denomina acceso anónimo, puede establecer el carácter comodín (`"*"`) como valor de `Principal`. Por ejemplo, si configura el bucket como sitio web, es porque desea que todos los objetos en el bucket tengan acceso público.

```
"Principal":"*"
```

```
"Principal":{"AWS":"*"}
```

Utilizar `"Principal": "*"` con un efecto de `Allow` en una política basada en recursos permite que cualquier persona, incluso si no ha iniciado sesión en AWS, acceda a su recurso.

Utilizar `"Principal" : { "AWS" : "*" }` con un efecto de `Allow` en una política basada en recursos permite que cualquier usuario raíz, usuario de IAM, sesión de rol asumido o usuario federado en cualquier cuenta de la misma partición acceda a su recurso.

Para los usuarios anónimos, estos dos métodos son equivalentes. Para obtener más información, consulte Todas las entidades principales en la Guía del usuario de IAM.

No puede utilizar un carácter comodín para buscar coincidencias con parte de un nombre de entidad principal o ARN.

> **Important**
>
> Puesto que cualquiera puede crear una Cuenta de AWS, el nivel de seguridad de estos dos métodos es equivalente, aunque funcionan de forma diferente.

Restringir los permisos de recursos

También puede utilizar la política de recursos para restringir el acceso a los recursos que, de otro modo, estarían disponibles para las entidades principales de IAM. Utilice una instrucción `Deny` para impedir el acceso.

El siguiente ejemplo bloquea el acceso si no se utiliza un protocolo de transporte seguro:

```
{"Effect": "Deny",
 "Principal": "*",
 "Action": "s3:*",
 "Resource": <bucket ARN>,
 "Condition": {
   "Boolean": { "aws:SecureTransport" : "false"}
 }
}
```

Una práctica recomendada para esta política es usar `"Principal": "*"` para que esta restricción se aplique a todo el mundo, en lugar de intentar denegar el acceso solo a cuentas o entidades principales específicas mediante este método.

Requerir acceso a través de URL de CloudFront

Puede exigir a los usuarios que obtengan acceso al contenido de Amazon S3 solo mediante direcciones URL de CloudFront en lugar de direcciones URL de Amazon S3. Para ello, cree un control de acceso de origen (OAC) de CloudFront. A continuación, cambie los permisos de los datos de S3. En la política de bucket, puede configurar CloudFront como entidad principal de la siguiente manera:

```
"Principal":{"Service":"cloudfront.amazonaws.com"}
```

Utilice un elemento `Condition` en la política para permitir que CloudFront acceda al bucket solo cuando la solicitud sea en nombre de la distribución de CloudFront que contiene el origen de S3.

```
       "Condition": {
          "StringEquals": {
             "AWS:SourceArn":
 "arn:aws:cloudfront::111122223333:distribution/CloudFront-distribution-ID"
          }
```

```
        }
```

Para obtener más información acerca de cómo solicitar acceso de S3 a través de las URL de CloudFront, consulte Restricción del acceso a un origen de Amazon Simple Storage Service en la Guía para desarrolladores de Amazon CloudFront. Para obtener más información sobre los beneficios de seguridad y privacidad de usar Amazon CloudFront, consulte Configuración de acceso seguro y acceso restringido al contenido.

Ejemplos de políticas basadas en recursos para Amazon S3

- Para ver ejemplos de políticas de buckets de Amazon S3, consulte Políticas de buckets para Amazon S3.
- Para ver ejemplos de políticas de puntos de acceso, consulte Configurar las políticas de IAM para el uso de puntos de acceso.

Acciones de políticas para Amazon S3

Compatibilidad con las acciones de política: sí

Los administradores pueden utilizar las políticas JSON de AWS para especificar quién tiene acceso a qué. Es decir, qué entidad principal puede realizar acciones en qué recursos y en qué condiciones.

El elemento `Action` de una política JSON describe las acciones que puede utilizar para conceder o denegar el acceso en una política. Las acciones de la política generalmente tienen el mismo nombre que la operación de API de AWS asociada. Hay algunas excepciones, como acciones de solo permiso que no tienen una operación de API coincidente. También hay algunas operaciones que requieren varias acciones en una política. Estas acciones adicionales se denominan acciones dependientes.

Incluya acciones en una política para conceder permisos y así llevar a cabo la operación asociada.

A continuación, se muestran los diferentes tipos de relaciones de mapeo entre las operaciones de la API de S3 y las acciones de políticas requeridas.

- Mapeo uno a uno con el mismo nombre. Por ejemplo, para utilizar la operación de la API `PutBucketPolicy`, se necesita la acción de política `s3:PutBucketPolicy`.
- Mapeo uno a uno con diferentes nombres. Por ejemplo, para utilizar la operación de la API `ListObjectsV2`, se necesita la acción de política `s3:ListBucket`.

- Mapeo uno a varios. Por ejemplo, para utilizar la operación de la API `HeadObject`, se necesita `s3:GetObject`. Además, si utiliza el bloqueo de objetos de S3 y desea obtener el estado de retención legal o la configuración de retención de un objeto, también son necesarias las acciones de políticas `s3:GetObjectLegalHold` y `s3:GetObjectRetention` correspondientes para poder utilizar la operación de API `HeadObject`.
- Mapeo uno a varios. Por ejemplo, para utilizar las operaciones de la API `ListObjectsV2` o `HeadBucket`, se necesita la acción de política `s3:ListBucket`.

Para ver una lista de las acciones de Amazon S3 para usar en políticas, consulte Acciones definidas por Amazon S3 en la Referencia de autorizaciones de servicio. Para obtener una lista de operaciones de la API de Amazon S3, consulte Amazon S3 API Actions en la Referencia de la API de Amazon Simple Storage Service.

Las acciones de políticas de Amazon S3 utilizan el siguiente prefijo antes de la acción:

```
s3
```

Para especificar varias acciones en una única instrucción, sepárelas con comas.

```
"Action": [
      "s3:action1",
      "s3:action2"
         ]
```

Operaciones con buckets

Las operaciones de buckets son operaciones de la API de S3 que funcionan en el tipo de recurso de bucket. Por ejemplo, `CreateBucket`, `ListObjectsV2` y `PutBucketPolicy`. Las acciones de las políticas de S3 para operaciones de buckets requieren que el elemento `Resource` de las políticas de buckets o de las políticas basadas en identidades de IAM sea el identificador de Nombre de recurso de Amazon (ARN) del tipo de bucket de S3 en el siguiente formato de ejemplo.

```
"Resource": "arn:aws:s3:::amzn-s3-demo-bucket"
```

La siguiente política de buckets otorga al usuario *Akua* con la cuenta *12345678901* el permiso `s3:ListBucket` para realizar la operación de la API ListObjectsV2 y enumerar los objetos de un bucket de 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:::amzn-s3-demo-bucket"
        }
    ]
}
```

Operaciones de buckets en políticas de puntos de acceso

Los permisos concedidos en una política de puntos de acceso solo se aplican si el bucket subyacente permite los mismos permisos. Cuando utilice puntos de acceso de S3, debe delegar el control de acceso del bucket al punto de acceso o añadir los mismos permisos en las políticas del punto de acceso a la política del bucket subyacente. Para obtener más información, consulte Configurar las políticas de IAM para el uso de puntos de acceso. En las políticas de puntos de acceso, en las acciones de la política de S3 para las operaciones de buckets es necesario utilizar el ARN de `accesspoint` para el elemento `Resource` en el siguiente formato.

```
"Resource": "arn:aws:s3:us-west-2:123456789012:accesspoint/DOC-EXAMPLE-ACCESS-POINT"
```

La siguiente política de puntos de acceso otorga al usuario *Akua* con la cuenta *12345678901* el permiso `s3:ListBucket` para realizar la operación de la API ListObjectSv2 a través del punto de acceso de S3 *DOC-EXAMPLE-ACCESS-POINT* para enumerar los objetos del bucket asociado al punto de acceso.

```
{
    "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**
>
> El punto de acceso de S3 no admite todas las operaciones de bucket. Para obtener más información, consulte Compatibilidad de los puntos de acceso con las operaciones de S3.

Operaciones con objetos

Las operaciones de objetos son operaciones de la API de S3 que actúan en función del tipo de recurso del objeto. Por ejemplo, `GetObject`, `PutObject` y `DeleteObject`. Las acciones de políticas de S3 para las operaciones de objetos necesitan que el elemento `Resource` de las políticas sea el ARN del objeto de S3 en los siguientes formatos de ejemplo.

```
"Resource": "arn:aws:s3:::amzn-s3-demo-bucket/*"
```

```
"Resource": "arn:aws:s3:::amzn-s3-demo-bucket/prefix/*"
```

> **Note**
>
> El ARN del objeto debe contener una barra inclinada después del nombre del bucket, como se ha visto en los ejemplos anteriores.

La siguiente política de buckets concede al usuario *Akua* con la cuenta *12345678901* el permiso `s3:PutObject` para realizar la operación de la API PutObject para subir objetos a un bucket de 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:::amzn-s3-demo-bucket/*"
        }
    ]
}
```

Operaciones de objetos en políticas de puntos de acceso

Cuando utilice puntos de acceso de S3 para controlar el acceso a las operaciones de los objetos, puede utilizar políticas de puntos de acceso. Cuando utilice políticas de puntos de acceso, las acciones de la política de S3 para operaciones con objetos requieren que utilice el ARN `accesspoint` para el elemento `Resource` en el siguiente formato: `arn:aws:s3:region:account-id:accesspoint/access-point-name/object/resource`. Para las operaciones de objetos que utilizan el punto de acceso, debe incluir el valor `/object/` después de todo el ARN del punto de acceso en el elemento `Resource`. Estos son algunos ejemplos.

```
"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 siguiente política de puntos de acceso otorga al usuario *Akua* con la cuenta *12345678901* el permiso `s3:GetObject` para realizar la operación de la API GetObject a través del punto de acceso *DOC-EXAMPLE-ACCESS-POINT* en todos los objetos del bucket asociado al punto de acceso.

```
{
    "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**
>
> El punto de acceso de S3 no admite todas las operaciones de objetos. Para obtener más información, consulte Compatibilidad de los puntos de acceso con las operaciones de S3.

Operaciones de puntos de acceso

Las operaciones de puntos de acceso son operaciones de la API de S3 que funcionan en el tipo de recurso `accesspoint`. Por ejemplo, `CreateAccessPoint`, `DeleteAccessPoint` y `GetAccessPointPolicy`. Las acciones de políticas de S3 para las operaciones de punto de acceso solo pueden utilizarse en las políticas de IAM basadas en identidades, no en las políticas de buckets ni en las de puntos de acceso. Las operaciones de puntos de acceso requieren que el elemento `Resource` sea el ARN de `accesspoint` en el siguiente formato de ejemplo.

```
"Resource": "arn:aws:s3:us-west-2:123456789012:accesspoint/DOC-EXAMPLE-ACCESS-POINT"
```

La siguiente política de IAM basada en identidades concede el permiso `s3:GetAccessPointPolicy` para realizar la operación de la API GetAccessPointPolicy en el punto de acceso de 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"
        }
    ]
}
```

Cuando utilice puntos de acceso, para controlar el acceso a las operaciones de buckets, consulte Operaciones de buckets en políticas de puntos de acceso; para controlar el acceso a las operaciones de objetos, consulte Operaciones de objetos en políticas de puntos de acceso. Para obtener más información acerca de cómo configurar políticas de punto de acceso, consulte Configurar las políticas de IAM para el uso de puntos de acceso.

Operaciones de punto de acceso de Object Lambda

Con Amazon S3 Object Lambda, puede agregar su propio código a las solicitudes `GET`, `LIST` y `HEAD` de Amazon S3 para modificar y procesar los datos a medida que vuelven a una aplicación. Puede realizar solicitudes a través de un punto de acceso de Object Lambda, que funciona como realizar solicitudes a través de otros puntos de acceso. Para obtener más información, consulte Transformación de objetos con Lambda para objetos S3.

Para obtener más información sobre cómo configurar políticas para operaciones de puntos de acceso de Object Lambda, consulte Configuración de las políticas de IAM para puntos de acceso de Object Lambda.

Operaciones de puntos de acceso multirregión

Un punto de acceso de varias regiones proporciona un punto de conexión global que las aplicaciones pueden utilizar para satisfacer las solicitudes de los buckets de S3 ubicados en varias Región de AWS. Puede utilizar un punto de acceso de varias regiones para crear aplicaciones de varias regiones con la misma arquitectura que se utiliza en una sola región y, a continuación, ejecutar esas aplicaciones en cualquier parte del mundo. Para obtener más información, consulte Puntos de acceso de varias regiones de Amazon S3.

Para obtener más información sobre cómo configurar políticas para operaciones de puntos de acceso multiregionales, consulte Ejemplos de política de punto de acceso multirregional.

Operaciones de trabajos por lotes

Las operaciones de trabajos (operaciones por lotes) son operaciones de la API de S3 que funcionan en el tipo de recurso de trabajo. Por ejemplo, `DescribeJob` y `CreateJob`. Las acciones de políticas de S3 para operaciones de trabajo solo pueden utilizarse en políticas basadas en identidades de IAM, no en políticas de buckets. Además, las operaciones de trabajos requieren que el elemento `Resource` de las políticas basadas en identidades de IAM sean el ARN de `job` en el siguiente formato de ejemplo.

```
"Resource": "arn:aws:s3:*:123456789012:job/*"
```

La siguiente política de IAM basada en identidades concede el permiso `s3:DescribeJob` para realizar la operación de la API DescribeJob en el trabajo de operaciones por lotes *DOC-EXAMPLE-JOB* de S3.

```
{
    "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"
        }
    ]
}
```

Operaciones de configuración de S3 Storage Lens

Para obtener más información acerca de cómo configurar las operaciones de configuración de S3 Storage Lens, consulte Permisos de Amazon S3 Storage Lens.

Operaciones de cuentas

Las operaciones de cuentas son operaciones de la API de S3 que funcionan en el nivel de la cuenta. Por ejemplo, `GetPublicAccessBlock` (para una cuenta). La cuenta no es un tipo de recurso definido por Amazon S3. Las acciones de políticas de S3 para operaciones de cuentas solo pueden utilizarse en políticas basadas en identidades de IAM, no en políticas de buckets. Además,

las operaciones de cuentas requieren que el elemento `Resource` de las políticas basadas en identidades de IAM sean `"*"`.

La siguiente política de IAM basada en identidades concede el permiso `s3:GetAccountPublicAccessBlock` para realizar la operación de la API GetPublicAccessBlock en el nivel de la cuenta y recuperar la configuración del bloque de acceso público en el nivel de la cuenta.

```
{
   "Version":"2012-10-17",
   "Statement":[
      {
         "Sid":"Allow retrieving the account-level Public Access Block settings",
         "Effect":"Allow",
         "Action":[
            "s3:GetAccountPublicAccessBlock"
         ],
         "Resource":[
            "*"
         ]
       }
    ]
}
```

Ejemplos de políticas para Amazon S3

- Para ver ejemplos de políticas basadas en identidad de Amazon S3, consulte Políticas basadas en identidad para Amazon S3.
- Para ver ejemplos de políticas basadas en recursos de Amazon S3, consulte Políticas de buckets para Amazon S3 y Configurar las políticas de IAM para el uso de puntos de acceso.

Recursos de políticas para Amazon S3

Compatibilidad con los recursos de políticas: sí

Los administradores pueden utilizar las políticas JSON de AWS para especificar quién tiene acceso a qué. Es decir, qué entidad principal puede realizar acciones en qué recursos y en qué condiciones.

El elemento `Resource` de la política JSON especifica el objeto u objetos a los que se aplica la acción. Las instrucciones deben contener un elemento `Resource` o `NotResource`. Como práctica

recomendada, especifique un recurso utilizando el Nombre de recurso de Amazon (ARN). Puede hacerlo para acciones que admitan un tipo de recurso específico, conocido como permisos de nivel de recurso.

Para las acciones que no admiten permisos de nivel de recurso, como las operaciones de descripción, utilice un carácter comodín (*) para indicar que la instrucción se aplica a todos los recursos.

```
"Resource": "*"
```

Algunas acciones de la API de Amazon S3 admiten varios recursos. Por ejemplo, `s3:GetObject` accede a EXAMPLE-RESOURCE-1 y EXAMPLE-RESOURCE-2, por lo que una entidad principal debe tener permisos para acceder a ambos recursos. Para especificar varios recursos en una única instrucción, separe los ARN con comas.

```
"Resource": [
      "EXAMPLE-RESOURCE-1",
      "EXAMPLE-RESOURCE-2"
```

Los recursos de Amazon S3 son buckets, objetos, puntos de acceso o trabajos. En una política, utilice el nombre de recurso de Amazon (ARN) del bucket, objeto, punto de acceso o trabajo para identificar el recurso.

Para ver una lista completa de los tipos de recursos de Amazon S3 y los ARN, consulte Recursos definidos por Amazon S3 en la Referencia de autorizaciones de servicio. Para obtener información acerca de las acciones con las que puede especificar el ARN de cada recurso, consulte Acciones definidas por Amazon S3.

Comodines para los ARN de recursos

Puede utilizar comodines como parte del ARN del recurso. Puede utilizar caracteres comodín (* y ?) en cualquier segmento del ARN (las partes se separan mediante signos de dos puntos). Un asterisco (*) representa cualquier combinación de cero o más caracteres y un signo de interrogación (?) representa un único carácter. Puede utilizar varios caracteres * o ? en cada segmento, pero un carácter comodín no puede abarcar varios segmentos.

- El ARN siguiente utiliza el carácter comodín * en la parte de identificador relativo del ARN para identificar todos los objetos del bucket `examplebucket`.

```
arn:aws:s3:::examplebucket/*
```

- El siguiente ARN utiliza * para indicar todos los buckets de S3 y objetos.

```
arn:aws:s3:::*
```

- El siguiente ARN utiliza ambos comodines, * y ?, en la parte `relative-ID`. Identifica todos los objetos en los buckets como `example1bucket`, `example2bucket`, `example3bucket`, etc.

```
arn:aws:s3:::example?bucket/*
```

Variables de política para ARN de recursos

Puede utilizar variables de política en los ARN de Amazon S3. Cuando se evalúa una política, estas variables predefinidas se sustituyen por los valores correspondientes. Supongamos que organiza los buckets como una colección de carpetas, una carpeta para cada uno de los usuarios. El nombre de la carpeta será igual al nombre del usuario. Para conceder a los usuarios permisos para sus carpetas, puede especificar una variable de política en el ARN del recurso:

```
arn:aws:s3:::bucket_name/developers/${aws:username}/
```

En tiempo de ejecución, cuando se evalúa la política, la variable `${aws:username}` en el ARN del recurso se sustituye por el nombre de usuario de la persona que realiza la solicitud.

Ejemplos de políticas para Amazon S3

- Para ver ejemplos de políticas basadas en identidad de Amazon S3, consulte Políticas basadas en identidad para Amazon S3.
- Para ver ejemplos de políticas basadas en recursos de Amazon S3, consulte Políticas de buckets para Amazon S3 y Configurar las políticas de IAM para el uso de puntos de acceso.

Claves de condición de políticas para Amazon S3

Compatibilidad con claves de condición de políticas específicas del servicio: sí

Los administradores pueden utilizar las políticas JSON de AWS para especificar quién tiene acceso a qué. Es decir, qué entidad principal puede realizar acciones en qué recursos y en qué condiciones.

El elemento `Condition` (o bloque de `Condition`) permite especificar condiciones en las que entra en vigor una instrucción. El elemento `Condition` es opcional. Puede crear expresiones condicionales que utilicen operadores de condición, tales como igual o menor que, para que la condición de la política coincida con los valores de la solicitud.

Si especifica varios elementos de `Condition` en una instrucción o varias claves en un único elemento de `Condition`, AWS las evalúa mediante una operación `AND` lógica. Si especifica varios valores para una única clave de condición, AWS evalúa la condición con una operación lógica `OR`. Se deben cumplir todas las condiciones antes de que se concedan los permisos de la instrucción.

También puede utilizar variables de marcador de posición al especificar condiciones. Por ejemplo, puede conceder un permiso de usuario de IAM para acceder a un recurso solo si está etiquetado con su nombre de usuario de IAM. Para más información, consulte Elementos de la política de IAM: variables y etiquetas en la Guía del usuario de IAM.

AWS admite claves de condición globales y claves de condición específicas del servicio. Para ver todas las claves de condición globales de AWS, consulte Claves de contexto de condición globales de AWSen la Guía del usuario de IAM.

Cada clave de condición de Amazon S3 se asigna al encabezado de solicitud con el mismo nombre permitido por la API en la que se puede establecer la condición. Las claves de condición específicas de Amazon S3 dictan el comportamiento de los encabezados de solicitudes del mismo nombre. Por ejemplo, la clave de condición `s3:VersionId` utilizada para conceder permisos condicionales para el permiso
`s3:GetObjectVersion`
define el comportamiento del parámetro de consulta `versionId` que establece en una solicitud GET Object.

Para ver una lista de las claves de condición de Amazon S3, consulte Claves de condición para Amazon S3 en la Referencia de autorizaciones de servicio. Para obtener más información acerca de las acciones y los recursos con los que puede utilizar una clave de condición, consulte Acciones definidas por Amazon S3.

Ejemplo: restricción de cargas de objetos a objetos con una clase de almacenamiento específica

Supongamos que la cuenta A, representada por el ID de cuenta 123456789012, posee un bucket. El administrador de la cuenta A desea restringir a Dave, un usuario en la cuenta A, para que solo pueda cargar objetos en el bucket que se almacenan con la clase de almacenamiento `STANDARD_IA`. Para restringir las cargas de objetos a una clase de almacenamiento específica, el administrador de

cuenta A puede utilizar la clave de condición `s3:x-amz-storage-class`, como se muestra en la siguiente política de bucket de ejemplo.

```
{
                "Version": "2012-10-17",
                "Statement": [
                  {
                    "Sid": "statement1",
                    "Effect": "Allow",
                    "Principal": {
                      "AWS": "arn:aws:iam::123456789012:user/Dave"
                    },
                    "Action": "s3:PutObject",
                    "Resource": "arn:aws:s3:::amzn-s3-demo-bucket1/*",
                    "Condition": {
                      "StringEquals": {
                        "s3:x-amz-storage-class": [
                          "STANDARD_IA"
                        ]
                      }
                    }
                  }
                ]
            }
```

En el ejemplo, el bloque `Condition` especifica la condición `StringEquals` que se aplica al par clave-valor especificado, `"s3:x-amz-acl":["public-read"]`. Existe un conjunto de claves predefinidas que puede usar para expresar una condición. En el ejemplo se utiliza la clave de condición `s3:x-amz-acl`. Esta condición requiere que el usuario incluya el encabezado `x-amz-acl` con el valor `public-read` en cada solicitud PUT Object.

Ejemplos de políticas para Amazon S3

- Para ver ejemplos de políticas basadas en identidad de Amazon S3, consulte Políticas basadas en identidad para Amazon S3.
- Para ver ejemplos de políticas basadas en recursos de Amazon S3, consulte Políticas de buckets para Amazon S3 y Configurar las políticas de IAM para el uso de puntos de acceso.

ACL en Amazon S3

Compatibilidad con ACL: sí

En Amazon S3, las listas de control de acceso (ACL) controlan qué Cuentas de AWS tienen permisos para acceder a un recurso. Las ACL son similares a las políticas basadas en recursos, aunque no utilizan el formato de documento de políticas JSON.

> **Important**
>
> La mayoría de los casos de uso modernos de Amazon S3 ya no requieren el uso de ACL.

Para obtener información acerca de cómo usar ACL para controlar el acceso en Amazon S3, consulte Administración de acceso con ACL.

ABAC con Amazon S3

Compatibilidad con ABAC (etiquetas en las políticas): parcial

El control de acceso basado en atributos (ABAC) es una estrategia de autorización que define permisos en función de atributos. En AWS, estos atributos se denominan etiquetas. Puede adjuntar etiquetas a entidades de IAM (usuarios o roles) y a muchos recursos de AWS. El etiquetado de entidades y recursos es el primer paso de ABAC. A continuación, designa las políticas de ABAC para permitir operaciones cuando la etiqueta de la entidad principal coincida con la etiqueta del recurso al que se intenta acceder.

ABAC es útil en entornos que crecen con rapidez y ayuda en situaciones en las que la administración de las políticas resulta engorrosa.

Para controlar el acceso en función de etiquetas, debe proporcionar información de las etiquetas en el elemento de condición de una política utilizando las claves de condición `aws:ResourceTag/key-name`, `aws:RequestTag/key-name` o `aws:TagKeys`.

Si un servicio admite las tres claves de condición para cada tipo de recurso, el valor es Sí para el servicio. Si un servicio admite las tres claves de condición solo para algunos tipos de recursos, el valor es Parcial.

Para obtener más información sobre ABAC, consulte ¿Qué es ABAC? en la Guía del usuario de IAM. Para ver un tutorial con los pasos para configurar ABAC, consulte Uso del control de acceso basado en atributos (ABAC) en la Guía del usuario de IAM.

Para ver ejemplos de políticas basadas en identidades para limitar el acceso a trabajos de operaciones por lotes de S3 en función de etiquetas, consulte Control de permisos para Operaciones por lotes de S3 mediante etiquetas de trabajo.

ABAC y etiquetas de objetos

En las políticas de ABAC, los objetos utilizan etiquetas de `s3:` en lugar de etiquetas de `aws:`. Para controlar el acceso a objetos en función de etiquetas de objetos, debe proporcionar información de las etiquetas en el elemento de condición de una política con las siguientes etiquetas:

- `s3:ExistingObjectTag/tag-key`
- `s3:s3:RequestObjectTagKeys`
- `s3:RequestObjectTag/tag-key`

Para obtener información sobre el uso de etiquetas de objetos para controlar el acceso, incluidos ejemplos de políticas de permisos, consulte Etiquetado y políticas de control de acceso.

Uso de credenciales temporales con Amazon S3

Compatibilidad con credenciales temporales: sí

Algunos Servicios de AWS no funcionan cuando inicia sesión con credenciales temporales. Para obtener información adicional, incluida la información sobre qué Servicios de AWS funcionan con credenciales temporales, consulte Servicios de AWS que funcionan con IAM en la Guía del usuario de IAM.

Utiliza credenciales temporales si inicia sesión en la AWS Management Console con cualquier método, excepto un nombre de usuario y una contraseña. Por ejemplo, cuando accede a AWS utilizando el enlace de inicio de sesión único (SSO) de la empresa, ese proceso crea automáticamente credenciales temporales. También crea automáticamente credenciales temporales cuando inicia sesión en la consola como usuario y luego cambia de rol. Para más información sobre el cambio de roles, consulte Cambio a un rol (consola) en la Guía del usuario de IAM.

Puede crear credenciales temporales de forma manual mediante la AWS CLI o la API de AWS. A continuación, puede usar esas credenciales temporales para acceder a AWS. AWS recomienda generar credenciales temporales de forma dinámica en lugar de usar claves de acceso a largo plazo. Para más información, consulte Credenciales de seguridad temporales en IAM.

Sesiones de acceso directo para Amazon S3

Admite sesiones de acceso directo (FAS): sí

Cuando utiliza un usuario o un rol de IAM para llevar a cabo acciones en AWS, se lo considera una entidad principal. Cuando utiliza algunos servicios, es posible que realice una acción que

desencadene otra acción en un servicio diferente. FAS utiliza los permisos de la entidad principal para llamar a un Servicio de AWS, combinados con el Servicio de AWS solicitante para realizar solicitudes a servicios posteriores. Las solicitudes de FAS solo se realizan cuando un servicio recibe una solicitud que requiere interacciones con otros Servicios de AWS o recursos para completarse. En este caso, debe tener permisos para realizar ambas acciones. Para obtener información sobre las políticas a la hora de realizar solicitudes de FAS, consulte Reenviar sesiones de acceso.

- Amazon S3 utiliza FAS para realizar llamadas a AWS KMS para descifrar un objeto cuando se ha utilizado SSE-KMS para cifrarlo. Para obtener más información, consulte Uso del cifrado del servidor con claves de AWS KMS (SSE-KMS).
- Las concesiones de acceso de S3 también utilizan FAS. Tras crear una concesión de acceso a los datos de S3 para una identidad concreta, el concesionario solicita una credencial temporal de concesiones de acceso de S3. Las concesiones de acceso de S3 obtienen una credencial temporal para el solicitante de AWS STS y venden la credencial al solicitante. Para obtener más información, consulte Solicitar acceso a los datos de Amazon S3 a través de S3 Access Grants.

Roles de servicio para Amazon S3

Compatibilidad con roles de servicio: sí

Un rol de servicio es un rol de IAM que asume un servicio para realizar acciones en su nombre. Un administrador de IAM puede crear, modificar y eliminar un rol de servicio desde IAM. Para obtener más información, consulte Creación de un rol para delegar permisos a un Servicio de AWS en la Guía del usuario de IAM.

> **Warning**
>
> Es posible que cambiar los permisos de un rol de servicio interrumpa la funcionalidad de Amazon S3. Edite los roles de servicio solo cuando Amazon S3 proporcione orientación para hacerlo.

Roles vinculados a servicios para Amazon S3

Compatibilidad con roles vinculados al servicio: parcial

Un rol vinculado al servicio es un tipo de rol de servicio que está vinculado a un Servicio de AWS. El servicio puede asumir el rol para realizar una acción en su nombre. Los roles vinculados a servicios

aparecen en la Cuenta de AWS y son propiedad del servicio. Un administrador de IAM puede ver, pero no editar, los permisos de los roles vinculados a servicios.

Amazon S3 admite roles vinculados a servicios para Amazon S3 Storage Lens. Para obtener más información sobre cómo crear o administrar roles vinculados a servicios de Amazon S3, consulte Uso de roles vinculados a servicios para Amazon S3 Storage Lens.

Servicio de Amazon S3 como entidad principal

Nombre del servicio en la política	Característica de S3	Más información
`s3.amazonaws.com`	Replicación de S3	Configuración de la replicación en directo
`s3.amazonaws.com`	Notificaciones de eventos de S3	Notificaciones de eventos de Amazon S3
`s3.amazonaws.com`	Inventario de S3	Inventario de Amazon S3
`access-grants.s3.amazonaws.com`	Permisos de acceso de S3	Registrar una ubicación
`batchoperations.s3.amazonaws.com`	Operaciones por lotes de S3	Concesión de permisos para Operaciones por lotes de S3
`logging.s3.amazonaws.com`	Registro de acceso al servidor de S3	Habilitación del registro de acceso al servidor de Amazon S3
`storage-lens.s3.amazonaws.com`	Almacenamiento de lente de S3	Visualización de las métricas de Amazon S3 Storage Lens mediante una exportación de datos

Políticas y permisos en Amazon S3

Esta página proporciona una descripción general de las políticas de bucket y usuario en Amazon S3 y describe los elementos básicos de una política de AWS Identity and Access Management (IAM).

Los elementos que aparecen se vinculan a más detalles sobre ese elemento y ejemplos de cómo usarlo.

Para obtener una lista completa de las acciones, recursos y condiciones de Amazon S3, consulte Actions, resources, and condition keys for Amazon S3 en la Referencia de autorización de servicios.

Una política contiene los siguientes elementos básicos:

- Resource: el bucket, el objeto, el punto de acceso o el trabajo de Amazon S3 al que se aplica la política. Utilice el nombre de recurso de Amazon (ARN) del bucket, objeto, punto de acceso o trabajo para identificar el recurso.

  Un ejemplo de operaciones de nivel de bucket:

  `"Resource": "arn:aws:s3:::bucket_name"`

  Ejemplos de operaciones de nivel de objeto:

  - `"Resource": "arn:aws:s3:::bucket_name/*"` para todos los objetos del bucket.
  - `"Resource": "arn:aws:s3:::bucket_name/prefix/*"` para objetos con un determinado prefijo en el bucket.

  Para obtener más información, consulte Recursos de políticas para Amazon S3.

- Actions: Amazon S3 admite un conjunto de operaciones para cada recurso. Con las palabras clave de acción puede identificar las operaciones del recurso que desea permitir o denegar.

  Por ejemplo, el permiso `s3:ListBucket` autoriza al usuario a utilizar la operación ListObjectsV2 de Amazon S3. (El permiso `s3:ListBucket` es un caso en el que el nombre de la acción no se corresponde directamente con el nombre de la operación). Para obtener más información acerca del uso de acciones de Simple Storage Service (Amazon S3), consulte Acciones de políticas para Amazon S3. Para obtener una lista completa de las acciones de Amazon S3, consulte Acciones en la Referencia de la API de Amazon Simple Storage Service.

- Effect: el efecto obtenido cuando el usuario solicita la acción específica, que puede ser `Allow` o `Deny`.

  Si no concede acceso de forma explícita (permitir) a un recurso, el acceso se deniega implícitamente. También puede denegar explícitamente el acceso a un recurso. Es posible que haga esto para asegurarse de que un usuario no puede acceder al recurso, aunque una política

diferente le conceda acceso. Para obtener más información, consulte Elemento de la política de JSON de IAM: Efecto en la Guía del usuario IAM.

- Principal: la cuenta o el usuario con permiso de acceso a las acciones y los recursos en la instrucción. En una política de bucket, el principal es el usuario, cuenta, servicio u otra entidad que reciba este permiso. Para obtener más información, consulte Entidades principales de las políticas de bucket.
- Condition: condiciones para cuando se aplica una política. Puede utilizar claves generales de AWS y claves específicas de Amazon S3 para especificar condiciones en una política de acceso de Amazon S3. Para obtener más información, consulte Ejemplos de políticas de bucket que utilizan claves de condición.

En el siguiente ejemplo de política de bucket se muestran los elementos `Effect`, `Principal`, `Action` y `Resource`. Esta política permite que *Akua*, un usuario de la cuenta *123456789012*, tenga permisos `s3:GetObject`, `s3:GetBucketLocation` y `s3:ListBucket` de Amazon S3 en el bucket de *amzn-s3-demo-bucket1*.

```
{
    "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:::amzn-s3-demo-bucket1/*",
                "arn:aws:s3:::amzn-s3-demo-bucket1"
            ]
        }
    ]
}
```

Para obtener información completa acerca del lenguaje de las políticas, consulte Políticas y permisos en IAM y Referencia de políticas de JSON de IAM en la guía de usuario de IAM.

Delegación de permisos

Si una Cuenta de AWS es propietaria de un recurso, puede conceder esos permisos a otra Cuenta de AWS. Esa cuenta, a continuación, puede delegar esos permisos, o un subconjunto de ellos, a usuarios de la cuenta. Esto se conoce como delegación de permisos. Sin embargo, una cuenta que recibe permisos de otra cuenta no puede delegarlos a otra Cuenta de AWS con la modalidad entre cuentas.

Propiedad de los buckets y objetos de Amazon S3

Los buckets y los objetos son recursos de Amazon S3. De forma predeterminada, solo el propietario de los recursos puede obtener acceso a ellos. El propietario del recurso es la Cuenta de AWS que lo crea. Por ejemplo:

- La Cuenta de AWS que se usa para crear buckets y cargar objetos es la propietaria de dichos recursos.
- Si carga un objeto utilizando credenciales de usuario o rol de AWS Identity and Access Management (IAM), la Cuenta de AWS a la que pertenece el usuario o el rol es el propietario del objeto.
- Un propietario del bucket puede conceder permisos de una cuenta a otra Cuenta de AWS (o a usuarios de otra cuenta) para cargar objetos. En este caso, la Cuenta de AWS que carga los objetos es la propietaria. El propietario del bucket no tiene permisos sobre los objetos que son propiedad de otras cuentas, con las siguientes excepciones:
  - El propietario del bucket es quien paga las facturas. El propietario del bucket puede denegar el acceso a cualquier objeto, o eliminar cualquier objeto del bucket, independientemente de quién sea el propietario.
  - El propietario del bucket puede archivar los objetos o restaurar los objetos archivados, independientemente de quién sea el propietario. El archivado se refiere a la clase de almacenamiento empleada para almacenar los objetos. Para obtener más información, consulte Administración del ciclo de vida del almacenamiento.

Autenticación de la propiedad y las solicitudes

Todas las solicitudes a un bucket son autenticadas o no autenticadas. Las solicitudes autenticadas deben incluir un valor de firma que autentique al remitente de la solicitud y no es necesario en el

caso de las solicitudes no autenticadas. Para obtener más información acerca de la autenticación de solicitudes, consulte Realizar solicitudes.

El propietario del bucket puede permitir solicitudes no autenticadas. Por ejemplo, se permiten solicitudes PutObject no autenticadas cuando un bucket tiene una política de bucket pública o cuando una ACL de bucket concede acceso `WRITE` o `FULL_CONTROL` específicamente al grupo de `All Users` o al usuario anónimo. Para obtener más información acerca de las políticas de bucket públicas y las listas de control de acceso (ACL) públicas, consulte Qué significa "pública".

Todas las solicitudes no autenticadas las realiza el usuario anónimo. En las ACL, este usuario se representa mediante el ID de usuario canónico específico`65a011a29cdf8ec533ec3d1ccaae921c`. Si se carga un objeto en un bucket mediante una solicitud no autenticada, el usuario anónimo es el propietario del objeto. La ACL del objeto predeterminado concede acceso `FULL_CONTROL` al usuario anónimo como propietario del objeto. Por lo tanto, Amazon S3 permite solicitudes no autenticadas para recuperar el objeto o modificar su ACL.

Para impedir que el usuario anónimo modifique objetos, le recomendamos que no implemente políticas de bucket que permitan escrituras públicas anónimas en su bucket o que utilicen ACL que permitan al usuario anónimo acceso de escritura a su bucket. Para imponer este comportamiento recomendado, utilice Block Public Access de Amazon S3.

Para obtener más información acerca del bloqueo del acceso público, consulte Bloquear el acceso público a su almacenamiento de Amazon S3. Para obtener más información acerca de las ACL, consulte Información general de las Listas de control de acceso (ACL).

> **Important**
>
> Recomendamos que no utilice las credenciales del usuario raíz de la Cuenta de AWS para realizar solicitudes autenticadas. En su lugar, cree un rol de IAM y concédale derechos de acceso completos. Los usuarios con este rol se denominan usuarios administradores. Puede utilizar las credenciales asignadas al rol de administrador, en lugar de las credenciales de usuario raíz de la Cuenta de AWS, para interactuar con AWS y realizar tareas, como crear un bucket, crear usuarios y conceder permisos. Para obtener más información, consulte Credenciales de seguridad de AWS y Prácticas recomendadas de IAM en la Guía del usuario de IAM.

Políticas de buckets para Amazon S3

Una política de bucket está basada en recursos que puede utilizar para conceder permisos de acceso al bucket de Amazon S3 y a los objetos que contiene. Solo el propietario del bucket puede asociar una política a un bucket. Los permisos asociados a un bucket se aplican a todos los objetos del bucket que son propiedad de la cuenta de propietario del bucket. Estos permisos no se aplican a los objetos que pertenecen a otras Cuentas de AWS.

S3 Object Ownership es una configuración de nivel de bucket de Amazon S3 que puede usar para controlar la propiedad de los objetos que se cargan en el bucket y para activar o desactivar las listas de control de acceso (ACL). De forma predeterminada, Object Ownership se establece en la configuración impuesta por el propietario del bucket y todas las ACL están deshabilitadas. El propietario del bucket posee todos los objetos del bucket y administra el acceso a los datos de forma exclusiva mediante políticas.

Las políticas de buckets utilizan el lenguaje de política de AWS Identity and Access Management (IAM) basado en JSON. Puede utilizar directivas de bucket para agregar o denegar permisos para los objetos de un bucket. Las políticas de bucket pueden permitir o denegar solicitudes en función de los elementos de la política. Estos elementos incluyen el solicitante, las acciones de S3, los recursos y los aspectos o condiciones de la solicitud (como la dirección IP utilizada para realizar la solicitud).

Por ejemplo, puede crear una política de bucket que haga lo siguiente:

- Conceder permisos entre cuentas para cargar objetos en el bucket de S3
- Asegurarse de que usted, el propietario del bucket, tenga el control total de los objetos cargados

Para obtener más información, consulte Ejemplos de políticas de bucket de Amazon S3.

> **Important**
>
> No puede usar una política de bucket para evitar eliminaciones ni transiciones mediante una regla de S3 Lifecycle. Por ejemplo, aunque la política de bucket deniegue todas las acciones a todas las entidades principales, la configuración de S3 Lifecycle seguirá funcionando con normalidad.

Los temas de esta sección proporcionan ejemplos y muestran cómo agregar una política de bucket en la consola de S3. Para obtener más información sobre las políticas basadas en la identidad,

consulte Políticas basadas en identidad para Amazon S3. Para obtener información sobre el lenguaje de las políticas de bucket, consulte Políticas y permisos en Amazon S3.

Temas

- Agregar una política de bucket mediante la consola de Amazon S3
- Control del acceso desde puntos de enlace de la VPC con políticas de bucket
- Ejemplos de políticas de bucket de Amazon S3
- Ejemplos de políticas de bucket que utilizan claves de condición

Agregar una política de bucket mediante la consola de Amazon S3

Puede utilizar el Generador de políticas de AWS y la consola de Amazon S3 para agregar una nueva política de bucket o editar una existente. Una política de bucket es una política de AWS Identity and Access Management (IAM) basada en recursos. Puede agregar una política de bucket a un bucket para conceder a otras Cuentas de AWS o usuarios de IAM permisos de acceso al bucket y los objetos que contiene. Los permisos de objetos solo se aplican a aquellos objetos que cree el propietario del bucket. Para obtener más información acerca de las políticas de bucket, consulte Administración de identidades y accesos para Amazon S3.

Asegúrese de resolver advertencias de seguridad, errores, advertencias generales y sugerencias de AWS Identity and Access Management Access Analyzer antes de guardar la política. IAM Access Analyzer ejecuta verificaciones de política para validarla contra la Gramática de la política de IAM y las prácticas recomendadas. Estas verificaciones generan hallazgos y proporcionan recomendaciones procesables para ayudarlo a crear políticas funcionales y que se ajustan a las prácticas recomendadas de seguridad. Para obtener más información sobre la validación de políticas utilizando IAM Access Analyzer, consulte Validación de políticas de IAM Access Analyzer en la Guía del usuario de IAM. Para ver una lista de advertencias, errores y sugerencias que devuelve IAM Access Analyzer, consulte Referencia de verificación de políticas de IAM Access Analyzer.

Para obtener instrucciones sobre la solución de errores con una política, consulte Solucionar errores de acceso denegado (403 Prohibido) en Amazon S3.

Para crear o editar una política de bucket

1. Inicie sesión AWS Management Console Management Console y abra la consola de Amazon S3 en https://console.aws.amazon.com/s3/.
2. En el panel de navegación izquierdo, elija Instancias.

3. En la lista Buckets (Buckets), elija el nombre del bucket para el que desea crear una política de bucket o cuya política de bucket quiera editar.
4. Elija la pestaña Permisos.
5. En Política de bucket, elija Editar. Aparece la página Edit bucket policy (Editar política de bucket).
6. En la página Edit bucket policy (Editar política de bucket), lleve a cabo alguna de las siguientes operaciones:

   - Para ver ejemplos de políticas de bucket en la Guía del usuario de Amazon S3, elija Policy examples (Ejemplos de políticas).
   - Para generar una política automáticamente o editar JSON en la sección Policy (Política), elija Policy generator (Generador de políticas).

   Si elige Generador de políticas, se abre el generador de políticas de AWS en una ventana nueva.

   a. En la página Generador de políticas de AWS, para Seleccionar tipo de política, elija Política de bucket de S3.

   b. Agregue una instrucción ingresando la información en los campos proporcionados y, a continuación, elija Agregar declaración. Repita este paso para tantas instrucciones como desee agregar. Para obtener más información acerca de estos campos, consulte la Referencia de los elementos de las políticas de JSON de IAM en la Guía del usuario de IAM.

   > **Note**
   >
   > Para mayor comodidad, la página Editar la política del bucket muestra el ARN del bucket (Nombre de recurso de Amazon) actual encima del campo de texto Política. Puede copiar este ARN para utilizarlo en las instrucciones de la página Generador de políticas de AWS.

   c. Una vez que haya terminado de agregar instrucciones, elija Generar política.

   d. Copie el texto de la política generada, elija Cerrar y vuelva a la página Editar política de bucket en la consola de Amazon S3.

7. En el cuadro Política, edite la política existente o pegue la política de bucket desde el generador de políticas de AWS. Asegúrese de resolver advertencias de seguridad, errores, advertencias generales y sugerencias antes de guardar la política.

   > **Note**
   >
   > Las políticas de bucket tienen un límite de tamaño de 20 KB.

8. (Opcional) Elija Preview external access (Vista previa del acceso externo) en la esquina inferior derecha para obtener una vista previa de cómo la política nueva afecta al acceso público y entre cuentas al recurso. Antes de guardar la política, puede comprobar si introduce nuevos hallazgos de IAM Access Analyzer o resuelve las conclusiones existentes. Si no ve un analizador activo, elija Go to Access Analyzer (Ir a Access Analyzer) para crear un analizador de la cuenta en Access Analyzer de IAM. Para obtener más información, consulte Vista previa del acceso en la Guía del usuario de IAM.
9. Elija Save changes (Guardar cambios), que le devuelve a la pestaña Permissions (Permisos).

Control del acceso desde puntos de enlace de la VPC con políticas de bucket

Puede utilizar las políticas de bucket de Amazon S3 para controlar el acceso a los buckets desde puntos de conexión específicos de la nube privada virtual (VPC) o VPC específicas. Esta sección incluye ejemplos de políticas de bucket que se pueden utilizar para controlar el acceso al bucket de Amazon S3 desde puntos de conexión de VPC. Para obtener información acerca de cómo configurar los puntos de enlace de la VPC, consulte Puntos de conexión de la VPC en la guía del usuario de VPC.

Una VPC le permite lanzar recursos de AWS en una red virtual que haya definido. Un punto de conexión de VPC le permite crear una conexión privada entre la VPC y otro Servicio de AWS. Esta conexión privada no requiere acceso a través de Internet, de una conexión de red privada virtual (VPN), de una instancia NAT o de AWS Direct Connect.

Un punto de conexión de la VPC de Amazon S3 es una entidad lógica dentro de una VPC que permite la conectividad solo a Amazon S3. El punto de conexión de la VPC direcciona las solicitudes a Amazon S3 y direcciona las respuestas de vuelta a la VPC. Los puntos de conexión de la VPC solo cambian la forma en que se direccionan las solicitudes. Los nombres de DNS y los puntos de enlace públicos de Amazon S3 seguirán funcionando con los puntos de enlace de la VPC. Para obtener información importante acerca de cómo utilizar los puntos de conexión de VPC con Amazon S3,

consulte Puntos de conexión de puerta de enlace y Puntos de conexión de puerta de enlace para Amazon S3 en la Guía del usuario de VPC.

Los puntos de enlace de VPC para Amazon S3 tienen dos formas de controlar el acceso a los datos de Amazon S3:

- Puede controlar qué solicitudes, usuarios o grupos obtienen acceso a través de un punto de conexión de la VPC específico. Para obtener información acerca de este tipo de control de acceso, consulte Control del acceso a los puntos de conexión de VPC con políticas de puntos de conexión en la Guía del usuario de VPC.
- Puede controlar qué VPC o puntos de enlace de la VPC tienen acceso a sus buckets a través de las políticas de bucket de Amazon S3. Para ver ejemplos de este tipo de control de acceso de política de bucket, consulte los siguientes temas sobre restricción de acceso.

Temas

- Restricción del acceso a un punto de conexión de la VPC específico
- Restricción del acceso a una VPC específica

> **Important**
>
> Al aplicar las políticas de bucket de Amazon S3 para los puntos de conexión de VPC que se describen en esta sección, es posible que bloquee el acceso al bucket involuntariamente. Los permisos de bucket pensados para limitar el acceso del bucket a las conexiones procedente de su punto de conexión de la VPC pueden bloquear todas las conexiones al bucket. Para obtener información acerca de cómo corregir este problema, consulte ¿Cómo ajusto mi política de bucket cuando tiene la VPC o el ID de punto de conexión de VPC incorrectos? en el Centro de conocimiento de AWS Support.

Restricción del acceso a un punto de conexión de la VPC específico

El siguiente es un ejemplo de una política de bucket de Amazon S3 que restringe el acceso a un bucket específico, `awsexamplebucket1`, solo desde el punto de conexión de la VPC con el ID `vpce-1a2b3c4d`. Si el punto de conexión específico no se usa, la política deniega todo el acceso al bucket. La condición `aws:SourceVpce` especifica el punto de conexión. La condición `aws:SourceVpce` no requiere un nombre de recurso de Amazon (ARN) para el recurso de punto de conexión de VPC, solo el ID del punto de conexión de VPC. Para obtener más información acerca

del uso de las condiciones en una política, consulte Ejemplos de políticas de bucket que utilizan claves de condición.

> **Important**
>
> - Antes de utilizar la política de ejemplo siguiente, reemplace el ID del punto de conexión de la VPC por un valor adecuado para su caso de uso. De lo contrario, no podrá acceder a su bucket.
> - Esta política desactiva el acceso a la consola al bucket especificado, ya que las solicitudes de consola no se originan en el punto de conexión de VPC especificado.

```
{
   "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"
         }
       }
     }
   ]
}
```

Restricción del acceso a una VPC específica

Puede crear una política de bucket para restringir el acceso a una VPC específica con la condición `aws:SourceVpc`. Esto es útil si tiene múltiples puntos de enlace de la VPC configurados en la misma VPC y desea administrar el acceso a sus buckets de Amazon S3 para todos sus puntos de enlace. El siguiente es un ejemplo de una política que deniega el acceso a `awsexamplebucket1` y sus objetos desde cualquier VPC `vpc-111bbb22` exterior. Si la VPC especificada no se usa, la política deniega todo el acceso al bucket. Esta instrucción no concede acceso al bucket. Para

conceder el acceso, debe agregar una instrucción `Allow` independiente. La clave de condición `vpc-111bbb22` no requiere un ARN para el recurso de VPC, solo el ID de VPC.

> **Important**
>
> - Antes de utilizar la política de ejemplo siguiente, reemplace el ID de VPC por un valor adecuado para su caso de uso. De lo contrario, no podrá acceder a su bucket.
> - Esta política desactiva el acceso a la consola al bucket especificado porque las solicitudes de consola no se originan en la VPC especificada.

```
{
   "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"
         }
       }
     }
   ]
}
```

Ejemplos de políticas de bucket de Amazon S3

Con las políticas de bucket de Amazon S3, puede proteger el acceso a los objetos de los buckets, de modo que solo los usuarios con los permisos adecuados puedan acceder a ellos. Incluso puede impedir que los usuarios autenticados sin los permisos adecuados accedan a los recursos de Amazon S3.

En esta sección se presentan ejemplos de casos de uso típicos para políticas de bucket. Estas políticas de ejemplo usan *amzn-s3-demo-bucket* como valor de recurso. Para probar estas

políticas, sustituya *`user input placeholders`* por su propia información (como el nombre del bucket).

Para conceder o denegar permisos a un conjunto de objetos, puede usar caracteres comodín (*) en nombres de recurso de Amazon (ARN) y otros valores. Por ejemplo, puede controlar el acceso a grupos de objetos que empiezan por un prefijo común o terminar con una extensión específica, como `.html`.

Para obtener más información sobre el lenguaje de la política de AWS Identity and Access Management (IAM), consulte Políticas y permisos en Amazon S3.

> **Note**
>
> Si utiliza la consola de Amazon S3 para probar los permisos, debe conceder permisos adicionales requeridos por la consola: `s3:ListAllMyBuckets`, `s3:GetBucketLocation` y `s3:ListBucket`. Para ver un ejemplo de un tutorial en el que se conceden permisos a usuarios y se prueban esos permisos utilizando la consola, consulte Controlar el acceso a un bucket con las políticas de usuario.

Los recursos adicionales para crear políticas de bucket incluyen lo siguiente:

- Para obtener una lista de las acciones, recursos y claves de condición de política de IAM que se pueden utilizar al crear una política de buckets, consulte Acciones, recursos y claves de condición para Amazon S3 en la Referencia de autorizaciones de servicios.
- Para obtener información sobre cómo crear una política de S3, consulte Agregar una política de bucket mediante la consola de Amazon S3.
- Para solucionar los errores de una política, consulte Solucionar errores de acceso denegado (403 Prohibido) en Amazon S3.

Temas

- Concesión de permisos de solo lectura a un usuario anónimo público
- Requerir cifrado
- Administración de buckets mediante ACL predefinidas
- Administración del acceso a objetos con etiquetado de objetos
- Administración del acceso a objetos mediante claves de condición globales

- Administración del acceso en función de solicitudes HTTP o HTTPS
- Administración del acceso de los usuarios a carpetas específicas
- Administración del acceso para los registros de acceso
- Administración del acceso a una OAI de Amazon CloudFront
- Administración del acceso para la Lente de almacenamiento de Amazon S3
- Administración de permisos para inventario de S3, análisis de S3 e informes de inventario de S3
- Exigir MFA
- Prevención de que los usuarios eliminen objetos

Concesión de permisos de solo lectura a un usuario anónimo público

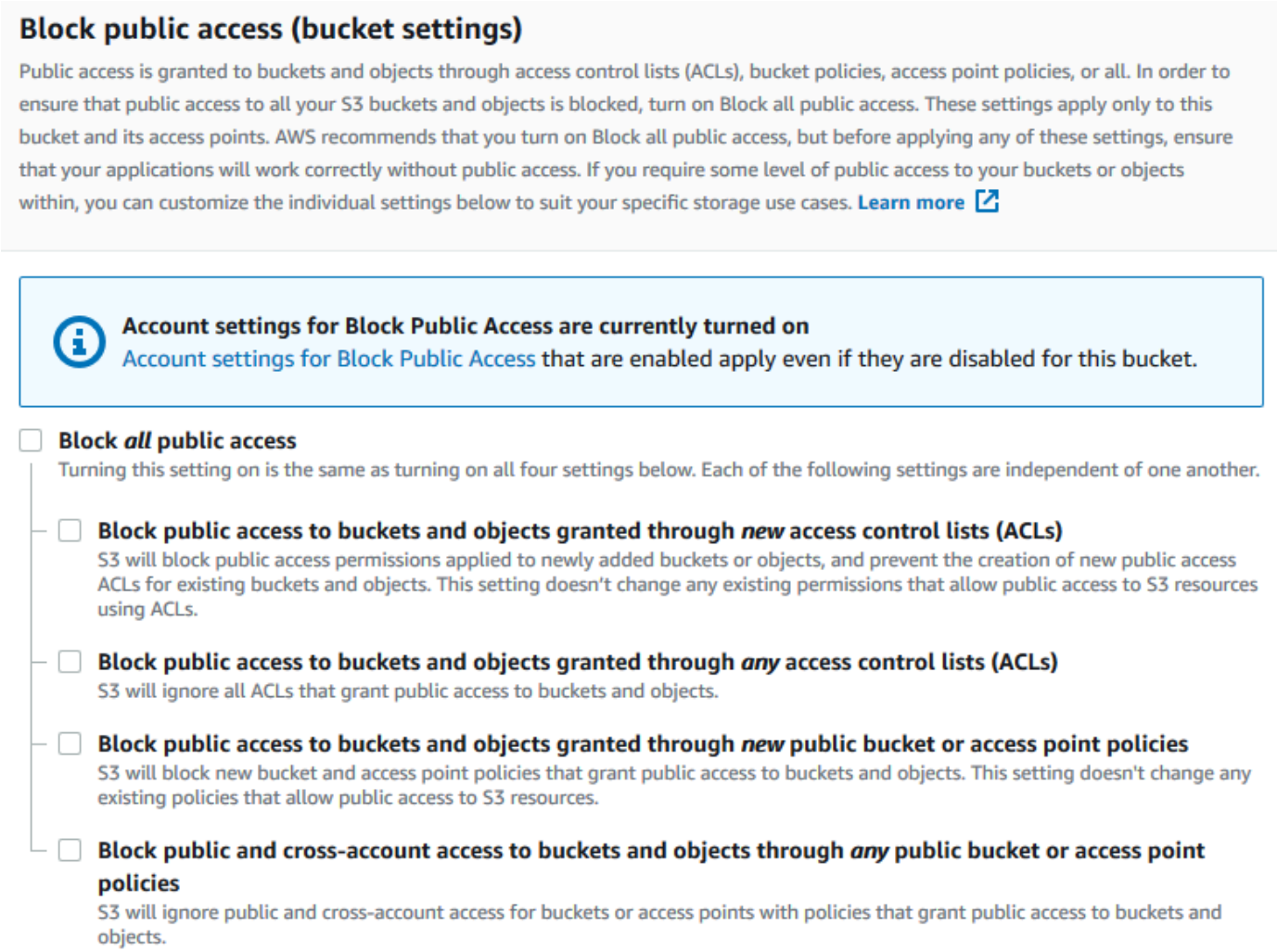
Puede usar la configuración de la política para conceder acceso a usuarios anónimos públicos, lo que resulta útil si está configurando el bucket como un sitio web estático. Para conceder acceso a usuarios públicos anónimos, es necesario deshabilitar la configuración de Bloqueo de acceso público para su bucket. Para obtener más información acerca de cómo hacerlo y la política necesaria, consulte Configurar permisos para el acceso a sitios web. Para obtener información sobre cómo configurar políticas más restrictivas con el mismo propósito, consulte ¿Cómo puedo conceder acceso de lectura público a algunos objetos de mi bucket de Amazon S3? en el Centro de conocimiento de AWS.

De forma predeterminada, Amazon S3 bloquea el acceso público a su cuenta y sus buckets. Si desea utilizar un bucket para alojar un sitio web estático, puede utilizar estos pasos para editar la configuración de bloqueo de acceso público.

> **Warning**
>
> Antes de completar estos pasos, consulte Bloquear el acceso público a su almacenamiento de Amazon S3 para asegurarse de que comprende y acepta los riesgos que implica permitir el acceso público. Cuando desactiva la configuración de acceso público de bloqueo para que el bucket sea público, cualquier usuario de Internet puede acceder al bucket. Le recomendamos que bloquee todo el acceso público a sus buckets.

1. Abra la consola de Amazon S3 en https://console.aws.amazon.com/s3.
2. Elija el nombre del bucket que ha configurado como sitio web estático.
3. Elija Permissions (Permisos).

4. En Block public access (bucket settings) (Bloquear acceso público [configuración de bucket]), elija Edit (Editar).
5. Desactive Block all public access (Bloquear todo el acceso público) y elija Save changes (Guardar cambios).



Amazon S3 desactiva la configuración de Bloqueo de acceso público para su bucket. Para crear un sitio web público y estático, es posible que también tenga que editar la configuración de Bloqueo de acceso público para su cuenta antes de agregar una política de bucket. Si la configuración de Bloqueo de acceso público de su cuenta está activada actualmente, verá una nota en Bloquear acceso público (configuración del bucket).

Requerir cifrado

Puede requerir cifrado del servidor con claves de AWS Key Management Service (AWS KMS) (SSE-KMS), como se muestra en los siguientes ejemplos.

Requerir SSE-KMS para todos los objetos escritos en un bucket

La política de ejemplo siguiente requiere que todos los objetos que se escriben en el bucket se cifren con el cifrado del lado del servidor mediante claves de AWS Key Management Service (AWS KMS) (SSE-KMS). Si el objeto no está cifrado con SSE-KMS, se deniega la solicitud.

```
{
"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"
    }
  }
}]
}
```

Requerir SSE-KMS con una AWS KMS key específica para todos los objetos escritos en un bucket

La siguiente política de ejemplo impide que se escriban objetos en el bucket si no están cifrados con SSE-KMS mediante un ID de clave KMS específico. Aunque los objetos se cifran con SSE-KMS mediante un cifrado predeterminado de encabezado por solicitud o de bucket, los objetos no se pueden escribir en el bucket si no se han cifrado con la clave KMS especificada. Asegúrese de sustituir el ARN de clave KMS que se usa en este ejemplo por su propio ARN de clave KMS.

```
{
"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"
    }
  }
}]
}
```

Administración de buckets mediante ACL predefinidas

Concesión de permisos a varias cuentas para cargar objetos o establecer ACL de objetos para el acceso público

La política de ejemplo siguiente concede permisos `s3:PutObject` y `s3:PutObjectAcl` a múltiples Cuentas de AWS. Además, la política de ejemplo requiere que cualquier solicitud de estas operaciones incluya la lista de control de acceso (ACL) predefinida `public-read`. Para obtener más información, consulte Acciones de políticas para Amazon S3 y Claves de condición de políticas para Amazon S3.

> **Warning**
>
> La `public-read` ACL predefinida permite que cualquier persona del mundo vea los objetos del bucket. Tenga cuidado al conceder acceso anónimo a su bucket de Amazon S3 o al deshabilitar la configuración del bloqueo de acceso público. Al otorgar acceso anónimo, cualquier persona puede acceder a su bucket. Le recomendamos que no conceda nunca acceso anónimo a su bucket de Amazon S3 a menos que lo necesite específicamente, por ejemplo con alojamiento de sitios web estáticos. Si desea habilitar la configuración del acceso público de bloques para el alojamiento de sitios web estáticos, consulte el Tutorial: Configuración de un sitio web estático en 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"
                    ]
                }
            }
        }
    ]
}
```

Conceder permisos entre cuentas para cargar objetos al mismo tiempo que se garantiza que el propietario del bucket tenga el control total

En el siguiente ejemplo, se muestra cómo permitir que otra Cuenta de AWS cargue objetos en el bucket al tiempo que se asegura de que tiene el control total de los objetos cargados. Esta política otorga una Cuenta de AWS específica (*111122223333*) la capacidad de cargar objetos solo si esa cuenta incluye la ACL predefinida `bucket-owner-full-control` en la carga. La condición `StringEquals` en la política especifica la clave de condición `s3:x-amz-acl` para expresar el requisito de ACL predefinido. Para obtener más información, consulte Claves de condición de políticas para 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"}
       }
     }
```

```
   ]
}
```

Administración del acceso a objetos con etiquetado de objetos

Permitir a un usuario leer solo los objetos que tienen una clave y valor de etiqueta específicos

La siguiente política de permisos limita al usuario a leer solo los objetos que tengan la clave y el valor de la etiqueta `environment: production`. Esta política usa la clave de condición `s3:ExistingObjectTag` para especificar la clave y el valor de etiqueta.

```
{
   "Version":"2012-10-17",
   "Statement":[
      {
         "Principal":{
            "AWS":"arn:aws:iam::111122223333:role/JohnDoe"
         },
         "Effect":"Allow",
         "Action":[
            "s3:GetObject",
            "s3:GetObjectVersion"
         ],
         "Resource":"arn:aws:s3:::amzn-s3-demo-bucket/*",
         "Condition":{
            "StringEquals":{
               "s3:ExistingObjectTag/environment":"production"
            }
         }
      }
   ]
}
```

Restringir las claves de etiqueta de objetos que los usuarios pueden agregar

La política de ejemplo siguiente concede un permiso de usuario para realizar la acción `s3:PutObjectTagging`, lo que permite al usuario agregar etiquetas a un objeto existente. La condición usa la clave de condición `s3:RequestObjectTagKeys` para especificar las claves de etiqueta permitidas, como `Owner` o `CreationDate`. Para obtener más información, consulte Creación de una condición que pruebe valores de varias claves en la Guía para usuarios de IAM.

La política garantiza que cada clave de etiqueta especificada en la solicitud sea una clave de etiqueta autorizada. El calificador `ForAnyValue` de la condición garantiza que al menos una de las claves especificadas estará presente en la solicitud.

```
{
   "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"
          ]
        }
      }
    }
  ]
}
```

Requerir una clave y un valor de etiqueta específica al permitir a los usuarios agregar etiquetas de objetos

La política de ejemplo siguiente concede un permiso de usuario para realizar la acción `s3:PutObjectTagging`, lo que permite al usuario agregar etiquetas a un objeto existente. La condición requiere que el usuario incluya una clave de etiqueta específica (como *`Project`*) con el valor establecido en *`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"
        }
      }
    }
  ]
}
```

Permitir a un usuario agregar objetos solo con una clave y valor de etiqueta de objetos específicos

La política de ejemplo siguiente concede un permiso de usuario para realizar la acción `s3:PutObject`, de modo que pueda agregar objetos a un bucket. Sin embargo, la instrucción `Condition` restringe las claves de etiqueta y los valores permitidos en los objetos cargados. En este ejemplo, el usuario solo puede agregar al bucket objetos que tengan la clave de etiqueta específica (*`Department`*) con el valor establecido en *`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"
            }
        }
```

```
    }]
}
```

Administración del acceso a objetos mediante claves de condición globales

Las claves de condición globales son claves de contexto de condición con un prefijo `aws`. Servicios de AWS pueden admitir claves de condición globales o proporcionar claves específicas del servicio que incluyan el prefijo de servicio. Puede utilizar el elemento `Condition` de una política JSON para comparar las claves de una solicitud con los valores de claves que especifique en la política.

Restringir el acceso solo a entregas de registros de acceso al servidor de Amazon S3

En la política de bucket de ejemplo siguiente, la clave de condición global aws:SourceArn se utiliza para comparar el Nombre de recurso de Amazon (ARN) del recurso, que realiza una solicitud de servicio a servicio con el ARN especificado en la política. La clave de condición global `aws:SourceArn` se usa para evitar que el servicio de Amazon S3 se utilice como un sustituto confuso durante las transacciones entre servicios. Solo el servicio de Amazon S3 tiene permiso para agregar objetos al bucket de Amazon S3.

Esta política de bucket de ejemplo concede permisos `s3:PutObject` solo a la entidad principal del servicio de registro (`logging.s3.amazonaws.com`).

> **Note**
>
> El elemento NotPrincipal no se puede utilizar con las entidades principales Servicio de AWS en las políticas basadas en recursos de Amazon S3, como las políticas de bucket. En su lugar, recomendamos utilizar la clave de condición `aws:PrincipalServiceName`, tal y como se muestra en la siguiente política.

```
{
    "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": {
                "StringNotEqualsIfExists": {
                    "aws:PrincipalServiceName": "logging.s3.amazonaws.com"
                }
            }
        }
    ]
}
```

Permitir el acceso solo a la organización

Si desea exigir que todas las entidades principales de IAM que accedan a un recurso provengan de una Cuenta de AWS en la organización (incluida la cuenta de administración de AWS Organizations), puede utilizar la clave de condición global `aws:PrincipalOrgID`.

Para conceder o restringir este tipo de acceso, defina la condición `aws:PrincipalOrgID` y establezca el valor del ID de la organización en la política de bucket. El ID de la organización se usa para controlar el acceso al bucket. Al usar la condición `aws:PrincipalOrgID`, los permisos de la política de bucket también se aplican a todas las cuentas nuevas que se agreguen a la organización.

Este es un ejemplo de política de bucket basada en recursos que puede utilizar para conceder a determinadas entidades principales de IAM de la organización acceso directo al bucket. Al agregar la clave de condición global `aws:PrincipalOrgID` a la política de bucket, ahora es necesario que la cuenta de la entidad principal esté en la organización para poder acceder al recurso. Aunque especifique accidentalmente una cuenta incorrecta al conceder el acceso, la clave de condición global aws:PrincipalOrgID actúa como una protección adicional. Cuando esta clave global se utiliza en una política, evita que todas las entidades principales de fuera de la organización especificada

accedan al bucket de S3. Solo las entidades principales de las cuentas de la organización mostradas pueden obtener acceso al recurso.

```
{
    "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"]
            }
        }
    }]
}
```

Administración del acceso en función de solicitudes HTTP o HTTPS

Restringir el acceso solo a las solicitudes HTTPS

Si quiere evitar que los posibles atacantes manipulen el tráfico de la red, puede usar HTTPS (TLS) para permitir solo conexiones cifradas y, al mismo tiempo, restringir el acceso de las solicitudes HTTP al bucket. Para determinar si la solicitud es HTTP o HTTPS, utilice la clave de condición global aws:SecureTransport en la política de bucket de S3. La clave de condición `aws:SecureTransport` comprueba si una solicitud se envió mediante HTTP.

Si una solicitud devuelve `true`, significa que la solicitud se envió a través de HTTPS. Si la solicitud devuelve `false`, significa que la solicitud se envió a través de HTTP. A continuación, puede permitir o denegar el acceso al bucket en función del esquema de solicitud deseado.

En el ejemplo siguiente, la política de bucket deniega explícitamente las solicitudes HTTP.

```
{
    "Version": "2012-10-17",
    "Statement": [{
        "Sid": "RestrictToTLSRequestsOnly",
        "Action": "s3:*",
```

```
        "Effect": "Deny",
        "Resource": [
            "arn:aws:s3:::DOC-EXAMPLE-BUCKET",
            "arn:aws:s3:::amzn-s3-demo-bucket/*"
        ],
        "Condition": {
            "Bool": {
                "aws:SecureTransport": "false"
            }
        },
        "Principal": "*"
    }]
}
```

Restringir el acceso a un HTTP referer específico

Suponga que tiene un sitio web con el nombre de dominio *www.example.com* o *example.com* con enlaces a fotografías y vídeos almacenados en el bucket denominado *amzn-s3-demo-bucket*. De forma predeterminada, todos los recursos de Amazon S3 son privados; por lo tanto, solo la Cuenta de AWS que creó los recursos puede acceder a ellos.

Para permitir el acceso de lectura a estos objetos desde el sitio web, puede agregar una política de bucket que conceda el permiso `s3:GetObject` con una condición de la solicitud `GET` que debe proceder de páginas web específicas. La política siguiente restringe las solicitudes mediante la condición `StringLike` con la clave de condición `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:::amzn-s3-demo-bucket/*",
      "Condition":{
        "StringLike":{"aws:Referer":["http://www.example.com/*","http://example.com/
*"]}
      }
    }
```

```
  ]
}
```

Asegúrese de que los navegadores que utiliza incluyan el encabezado HTTP `referer` en la solicitud.

> **Warning**
>
> Le recomendamos que actúe con precaución cuando utilice la clave de condición de `aws:Referer`. Es peligroso incluir un valor de encabezado de referencia HTTP conocido públicamente. Las partes no autorizadas podrían utilizar navegadores personalizados o modificados para proporcionar cualquier valor `aws:Referer` que eligieran. Por lo tanto, no utilice `aws:Referer` para evitar que las partes no autorizadas realicen solicitudes de AWS de forma directa.
>
> La clave de condición de `aws:Referer` se ofrece solo para que los clientes puedan proteger su contenido digital, como el contenido almacenado en Amazon S3, para evitar las referencias en sitios de terceros no autorizados. Para obtener más información, consulte aws:Referer en la Guía del usuario de IAM.

Administración del acceso de los usuarios a carpetas específicas

Conceder acceso a los usuarios a carpetas específicas

Suponga que está intentando conceder a los usuarios acceso a una carpeta específica. Si el usuario de IAM y el bucket de S3 pertenecen a la misma Cuenta de AWS, puede utilizar una política de IAM para conceder al usuario acceso a una carpeta de bucket específica. Con este enfoque, no es necesario actualizar la política de bucket para conceder el acceso. Puede agregar la política de IAM a un rol de IAM al que puedan cambiarse varios usuarios.

Si la identidad de IAM y el bucket de S3 pertenecen a Cuentas de AWS diferentes, debe conceder acceso entre cuentas tanto en la política de IAM como en la política de bucket. Para obtener más información sobre cómo conceder acceso entre cuentas, consulte Propietario del bucket que concede permisos de bucket entre cuentas.

La política de bucket de ejemplo siguiente concede a *JohnDoe* acceso completo a la consola solo a su carpeta (`home/`*JohnDoe*`/`). Al crear una carpeta `home` y conceder los permisos adecuados a los usuarios, puede hacer que varios usuarios compartan un solo bucket. Esta política consta de tres instrucciones `Allow`:

- *AllowRootAndHomeListingOfCompanyBucket*: permite al usuario (*JohnDoe*) mostrar los objetos en el nivel raíz del bucket *DOC-EXAMPLE-BUCKET* y en la carpeta home. Esta instrucción también permite al usuario buscar el prefijo home/ mediante la consola.
- *AllowListingOfUserFolder*: permite al usuario (*JohnDoe*) mostrar todos los objetos de la carpeta home/*JohnDoe*/ y cualquier subcarpeta.
- *AllowAllS3ActionsInUserFolder*: permite al usuario realizar todas las acciones de Amazon S3 mediante la concesión de los permisos `Read`, `Write` y `Delete`. Los permisos están limitados a la carpeta principal del propietario del bucket.

```
{
    "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/*"]
        }
    ]
}
```

Administración del acceso para los registros de acceso

Conceder acceso al equilibrador de carga de aplicación para habilitar los registros de acceso

Al habilitar los registros de acceso del equilibrador de carga de aplicación, debe especificar el nombre del bucket de S3 donde el equilibrador de carga almacenará los registros. El bucket debe tener una política asociada que conceda permiso a Elastic Load Balancing para escribir en el bucket.

En el siguiente ejemplo, la política de bucket concede permiso a Elastic Load Balancing (ELB) para escribir los registros de acceso en el bucket:

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Principal": {
        "AWS": "arn:aws:iam::elb-account-id:root"
      },
      "Effect": "Allow",
      "Action": "s3:PutObject",
      "Resource": "arn:aws:s3:::amzn-s3-demo-bucket/prefix/AWSLogs/111122223333/*"
    }
  ]
}
```

> **Note**
>
> Asegúrese de sustituir *elb-account-id* por el ID de Cuenta de AWS para Elastic Load Balancing para la Región de AWS. Para ver la lista de regiones de Elastic Load Balancing, consulte Adjuntar una política al bucket de Amazon S3 en la Guía del usuario de Elastic Load Balancing.

Si la Región de AWS no aparece en la lista de regiones de Elastic Load Balancing compatibles, utilice la siguiente política, que concede permisos al servicio de entrega de registros especificado.

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
       "Principal": {
         "Service": "logdelivery.elasticloadbalancing.amazonaws.com"
          },
      "Effect": "Allow",
      "Action": "s3:PutObject",
      "Resource": "arn:aws:s3:::amzn-s3-demo-bucket/prefix/AWSLogs/111122223333/*"
    }
  ]
}
```

A continuación, asegúrese de configurar los registros de acceso de Elastic Load Balancing habilitándolos. Puede verificar los permisos del bucket creando un archivo de prueba.

Administración del acceso a una OAI de Amazon CloudFront

Conceder permiso a una OAI de Amazon CloudFront

En el siguiente ejemplo de política de buckets, se concede un permiso de identidad de acceso de origen (OAI) de CloudFront para obtener (leer) todos los objetos del bucket de S3. Puede utilizar un OAI de CloudFront para permitir a los usuarios acceder a objetos del bucket a través de CloudFront, pero no directamente a través de Amazon S3. Para obtener más información, consulte Restricción del acceso a contenido de Amazon S3 utilizando una identidad de acceso de origen en la guía para desarrolladores de Amazon CloudFront.

La política siguiente utiliza el ID de la OAI como `Principal` de la política. Para obtener más información sobre el uso de políticas de bucket de S3 para conceder acceso a una OAI de

CloudFront, consulte Migración de la identidad de acceso de origen (OAI) al control de acceso de origen (OAC) en la Guía para desarrolladores de Amazon CloudFront.

Para usar este ejemplo:

- Reemplace *EH1HDMB1FH2TC* por el ID de la OAI. Para buscar el ID de la OAI, consulte la Origin Access Identity page (Página de identidad de acceso de origen) en la consola de CloudFront o utilice ListCloudFrontOriginAccessIdentities en la API de CloudFront.
- Reemplace *amzn-s3-demo-bucket* con el nombre de su bucket.

```
{
    "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:::amzn-s3-demo-bucket/*"
        }
    ]
}
```

Administración del acceso para la Lente de almacenamiento de Amazon S3

Conceder permisos para Lente de almacenamiento de Amazon S3

Lente de almacenamiento de S3 agrega las métricas y muestra la información en la sección Account snapshot (Instantánea de la cuenta) en la página Buckets de la consola de Amazon S3. S3 Storage Lens también proporciona un panel interactivo que puede utilizar para visualizar información y tendencias, marcar valores atípicos y recibir recomendaciones para optimizar los costes de almacenamiento y aplicar las prácticas recomendadas de protección de datos. El panel tiene opciones de desglose para generar y visualizar información en el nivel de la organización, la cuenta, la Región de AWS, la clase de almacenamiento, el bucket, el prefijo o el grupo de Lente de almacenamiento. También puede enviar una exportación de métricas diaria en formato CSV o Parquet a un bucket de S3.

Lente de almacenamiento de S3 puede exportar las métricas de uso de almacenamiento agregado a un bucket de Amazon S3 para un análisis posterior. El bucket en el que S3 Storage Lens coloca sus exportaciones de métricas se conoce como el bucket de destino. Al configurar la exportación de métricas de S3 Storage Lens, debe tener una política de buckets para el bucket de destino. Para obtener más información, consulte Evaluación de la actividad y el uso de almacenamiento con Amazon S3 Storage Lens.

En el siguiente ejemplo de política de bucket, se concede a Amazon S3 permiso para escribir objetos (solicitudes PUT) en un bucket de destino. Usted utiliza una política de bucket como esta en el bucket de destino cuando configura una exportación de métricas 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:::amzn-s3-demo-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"
                }
            }
        }
    ]
}
```

Cuando configure una exportación de métricas a nivel de organización de Lente de almacenamiento de S3, utilice la siguiente modificación de la instrucción `Resource` de la política de buckets anterior.

```
"Resource": "arn:aws:s3:::amzn-s3-demo-destination-bucket/destination-prefix/
StorageLens/your-organization-id/*",
```

Administración de permisos para inventario de S3, análisis de S3 e informes de inventario de S3

Concesión de permisos para el inventario de S3 y el análisis de S3

El inventario de S3 crea listas de los objetos en un bucket y la exportación de análisis de clase de almacenamiento de análisis de S3 crea archivos de salida de los datos utilizados en los análisis. El bucket para el que el inventario enumera los objetos se denomina bucket de origen. El bucket donde se escribe el archivo de inventario y el bucket donde se escribe se denomina bucket de destino. Al configurar un inventario o una exportación de análisis, debe crear una política de buckets para el bucket de destino. Para obtener más información, consulte Inventario de Amazon S3 y Análisis de Amazon S3: análisis de clases de almacenamiento.

En el siguiente ejemplo de política de bucket, se concede a Amazon S3 permiso para escribir objetos (solicitudes PUT) de la cuenta para el bucket de origen en el bucket de destino. Usted utiliza una política de bucket como esta en el bucket de destino cuando configura el inventario de S3 y la exportación de análisis de 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"
                }
```

```
            }
        }
    ]
}
```

Control de la creación de la configuración del informe de inventario de S3

Inventario de Amazon S3 crea listas de los objetos de un bucket de S3 y los metadatos para cada objeto. El permiso `s3:PutInventoryConfiguration` permite a un usuario crear una configuración de inventario que incluya todos los campos de metadatos de objeto disponibles de manera predeterminada y especificar el bucket de destino para almacenar el inventario. Un usuario con acceso de lectura a los objetos del bucket de destino puede acceder a todos los campos de metadatos de objetos que están disponibles en el informe de inventario. Para obtener más información acerca de los campos de metadatos disponibles en el inventario de S3, consulte Lista de Amazon S3 Inventory.

Para impedir que un usuario configure un informe de Inventario de S3, retire el permiso `s3:PutInventoryConfiguration` al usuario.

Algunos campos de metadatos de objetos en las configuraciones de los informes de Inventario de S3 son opcionales, lo que significa que están disponibles de manera predeterminada pero pueden restringirse cuando se concede el permiso `s3:PutInventoryConfiguration` a un usuario. Puede controlar si los usuarios pueden incluir estos campos de metadatos opcionales en sus informes mediante la clave de condición `s3:InventoryAccessibleOptionalFields`. Para obtener una lista de los campos de metadatos opcionales disponibles en Inventario de S3, consulte OptionalFields en la Referencia de la API de Amazon Simple Storage Service.

Para conceder a un usuario permiso para crear una configuración de inventario con campos de metadatos opcionales específicos, utilice la clave de condición `s3:InventoryAccessibleOptionalFields` para delimitar las condiciones de su política de buckets.

En el siguiente ejemplo de política, se otorga a un usuario (*Ana*) permiso para crear una configuración de inventario de forma condicional. En la condición `ForAllValues:StringEquals` de la política, se usa la clave de condición `s3:InventoryAccessibleOptionalFields` para especificar los dos campos de metadatos opcionales permitidos, es decir, `Size` y `StorageClass`. Por lo tanto, cuando *Ana* crea una configuración de inventario, los únicos campos de metadatos opcionales que puede incluir son `Size` y `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"
        ]
      }
    }
   }
 ]
}
```

Para evitar que un usuario configure un informe de Inventario de S3 que incluya campos de metadatos opcionales específicos, añada una instrucción `Deny` explícita a la política de buckets para el bucket de origen. En el siguiente ejemplo de política de buckets, se impide al usuario *Ana* crear una configuración de inventario en el bucket de origen *DOC-EXAMPLE-SOURCE-BUCKET* que incluya los campos de metadatos opcionales `ObjectAccessControlList` o `ObjectOwner`. El usuario *Ana* aún puede crear una configuración de inventario con otros campos de metadatos opcionales.

```
{
 "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

El uso de la clave de condición `s3:InventoryAccessibleOptionalFields` en las políticas de buckets no afecta a la entrega de informes de inventario basados en las configuraciones de inventario existentes.

Important

Recomendamos que use `ForAllValues` con un efecto `Allow` o `ForAnyValue` con un efecto `Deny`, tal y como se muestra en los ejemplos anteriores.
No utilice `ForAllValues` con un efecto `Deny` ni `ForAnyValue` con un efecto `Allow`, ya que estas combinaciones pueden ser demasiado restrictivas y bloquear la eliminación de la configuración del inventario.

Para obtener más información sobre los operadores de conjuntos de condiciones `ForAllValues` y `ForAnyValue`, consulte Claves de contexto multivalor en la Guía del usuario de IAM.

Exigir MFA

Amazon S3 admite el acceso de API protegido por MFA, una característica que puede exigir aplicar Multi-Factor Authentication (MFA) para acceder a sus recursos de Amazon S3. La autenticación multifactor proporciona un nivel de seguridad adicional que puede aplicar a su entorno de AWS. MFA es una característica de seguridad que requiere que los usuarios demuestren una posesión física de un dispositivo de MFA facilitando un código MFA válido. Para obtener más información, consulte autenticación multifactor de AWS. Puede solicitar MFA para cualquier solicitud de acceso a sus recursos de Amazon S3.

Para implementar el requisito de MFA, utilice la clave de condición `aws:MultiFactorAuthAge` en una política de bucket. Los usuarios de IAM pueden acceder a los recursos de Amazon S3 con credenciales temporales emitidas por AWS Security Token Service (AWS STS). Usted facilita el código de MFA al realizar la solicitud al AWS STS.

Cuando Amazon S3 recibe una solicitud con autenticación multifactor, la clave de condición `aws:MultiFactorAuthAge` proporciona un valor numérico que indica el tiempo que transcurrió (en segundos) desde que se creó la credencial temporal. Si la credencial temporal provista en la solicitud no se creó utilizando un dispositivo de MFA, este valor de clave es nulo (no está presente). En una política de bucket, puede añadir una condición para revisar este valor, como se muestra en el siguiente ejemplo.

Esta política de ejemplo deniega cualquier operación de Amazon S3 en la carpeta */taxdocuments* del bucket *amzn-s3-demo-bucket* si la solicitud no se autentica utilizando MFA. Para obtener más información sobre MFA, consulte Uso de autenticación multifactor (MFA) en AWS en la Guía del usuario de IAM.

```
{
    "Version": "2012-10-17",
    "Id": "123",
    "Statement": [
      {
        "Sid": "",
        "Effect": "Deny",
```

```
        "Principal": "*",
        "Action": "s3:*",
        "Resource": "arn:aws:s3:::amzn-s3-demo-bucket/taxdocuments/*",
        "Condition": { "Null": { "aws:MultiFactorAuthAge": true }}
      }
    ]
 }
```

La condición `Null` en el bloque `Condition` se evalúa como `true` si el valor de clave de condición `aws:MultiFactorAuthAge` es nulo, lo que indica que las credenciales de seguridad temporales en la solicitud se crearon sin un dispositivo de MFA.

La siguiente política de bucket es una extensión de la política de bucket anterior. La siguiente política incluye dos instrucciones de política. Una instrucción permite el permiso `s3:GetObject` en un bucket (*amzn-s3-demo-bucket*) para todo el mundo. Otra instrucción limita el acceso a la carpeta *amzn-s3-demo-bucket/taxdocuments* en el bucket mediante la solicitud de la MFA.

```
{
    "Version": "2012-10-17",
    "Id": "123",
    "Statement": [
      {
        "Sid": "",
        "Effect": "Deny",
        "Principal": "*",
        "Action": "s3:*",
        "Resource": "arn:aws:s3:::amzn-s3-demo-bucket/taxdocuments/*",
        "Condition": { "Null": { "aws:MultiFactorAuthAge": true } }
      },
      {
        "Sid": "",
        "Effect": "Allow",
        "Principal": "*",
        "Action": ["s3:GetObject"],
        "Resource": "arn:aws:s3:::amzn-s3-demo-bucket/*"
      }
    ]
 }
```

Puede utilizar de forma opcional una condición numérica para limitar la duración de la validez de la clave `aws:MultiFactorAuthAge`. La duración que especifique con la clave

`aws:MultiFactorAuthAge` es independiente de la duración de la credencial de seguridad temporal que se utiliza en la autenticación de la solicitud.

Por ejemplo, la siguiente política de bucket, además de exigir la autenticación MFA, también verifica el tiempo que transcurrió desde que se creó la sesión temporal. La política deniega cualquier operación si el valor de clave `aws:MultiFactorAuthAge` indica que la sesión temporal se creó hace más de una hora (3600 segundos).

```
{
    "Version": "2012-10-17",
    "Id": "123",
    "Statement": [
      {
        "Sid": "",
        "Effect": "Deny",
        "Principal": "*",
        "Action": "s3:*",
        "Resource": "arn:aws:s3:::amzn-s3-demo-bucket/taxdocuments/*",
        "Condition": {"Null": {"aws:MultiFactorAuthAge": true }}
      },
      {
        "Sid": "",
        "Effect": "Deny",
        "Principal": "*",
        "Action": "s3:*",
        "Resource": "arn:aws:s3:::amzn-s3-demo-bucket/taxdocuments/*",
        "Condition": {"NumericGreaterThan": {"aws:MultiFactorAuthAge": 3600 }}
       },
       {
         "Sid": "",
         "Effect": "Allow",
         "Principal": "*",
         "Action": ["s3:GetObject"],
         "Resource": "arn:aws:s3:::amzn-s3-demo-bucket/*"
       }
    ]
 }
```

Prevención de que los usuarios eliminen objetos

De forma predeterminada, los usuarios no tienen permisos. Sin embargo, a medida que crea políticas, puede conceder permisos a los usuarios que no quería conceder. Para evitar esas

ambigüedades en los permisos, puede escribir una política de acceso más estricta y agregar una denegación explícita.

Para impedir de forma explícita que los usuarios o las cuentas eliminen objetos, debe agregar las siguientes acciones a una política de bucket: permisos `s3:DeleteObject`, `s3:DeleteObjectVersion`, y `s3:PutLifecycleConfiguration`. Las tres acciones son necesarias porque puede eliminar objetos al llamar de forma explícita a las operaciones de la API `DELETE Object` o configurar su ciclo de vida (consulte Administración del ciclo de vida del almacenamiento) para que Amazon S3 pueda eliminar los objetos cuando acabe su vida útil.

En el siguiente ejemplo de política, debe denegar explícitamente permisos `DELETE Object` al usuario *`MaryMajor`*. Una instrucción `Deny` explícita siempre sustituye a cualquier otro permiso concedido.

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Sid": "statement1",
      "Effect": "Allow",
      "Principal": {
        "AWS": "arn:aws:iam::123456789012:user/MaryMajor"
      },
      "Action": [
        "s3:GetObjectVersion",
        "s3:GetBucketAcl"
      ],
      "Resource": [
        "arn:aws:s3:::amzn-s3-demo-bucket1",
   "arn:aws:s3:::amzn-s3-demo-bucket1/*"
      ]
    },
    {
      "Sid": "statement2",
      "Effect": "Deny",
      "Principal": {
        "AWS": "arn:aws:iam::123456789012:user/MaryMajor"
      },
      "Action": [
        "s3:DeleteObject",
        "s3:DeleteObjectVersion",
        "s3:PutLifecycleConfiguration"
```

```
      ],
      "Resource": [
        "arn:aws:s3:::amzn-s3-demo-bucket1",
     "arn:aws:s3:::amzn-s3-demo-bucket1/*"
      ]
    }
  ]
}
```

Ejemplos de políticas de bucket que utilizan claves de condición

Puede utilizar el lenguaje de la política de acceso para especificar condiciones al conceder permisos. Puede utilizar el elemento opcional `Condition` o el bloque `Condition` para especificar condiciones para cuando una política está en vigor.

Para obtener información sobre las políticas que utilizan claves de condición de Amazon S3 para operaciones de objetos y bucket, consulte los siguientes ejemplos. Para obtener más información acerca de las claves de condición, consulte Claves de condición de políticas para Amazon S3. Para obtener una lista completa de las acciones, claves de condición y recursos de Amazon S3 que puede especificar en las políticas, consulte Actions, resources, and condition keys for Amazon S3 en la Referencia de autorización de servicios.

Ejemplos: claves de condición de Amazon S3 para operaciones con objetos

Esta sección incluye ejemplos que muestran cómo puede utilizar claves de condición específicas de Amazon S3 para operaciones con objetos. Para obtener una lista completa de las acciones, claves de condición y recursos de Amazon S3 que puede especificar en las políticas, consulte Actions, resources, and condition keys for Amazon S3 en la Referencia de autorización de servicios.

Varias de las políticas de ejemplo muestran cómo se pueden utilizar las claves de condiciones con las operaciones PUT Object. La operación PUT Object permite encabezados específicos para la lista de control de acceso (ACL) que puede utilizar para conceder permisos basados en la ACL. Con estas claves, el propietario del bucket puede configurar una condición para solicitar permisos de acceso específicos cuando el usuario carga un objeto. También puede conceder permisos basados en ACL con la operación PutObjectAcl. Para obtener más información, consulte PutObjectAcl en la referencia de la API de Amazon Simple Storage Service de Amazon S3. Para obtener más información acerca de las ACL, consulte Información general de las Listas de control de acceso (ACL).

Temas

Ejemplo 1: Concesión de permisos s3:PutObject con una condición que requiere que los objetos almacenados tengan cifrado del servidor

Supongamos que la cuenta A tiene un bucket. El administrador de la cuenta desea conceder permisos para cargar objetos a Jane, usuario en la cuenta A, con la condición de que Jane siempre solicite el cifrado del lado del servidor para que Amazon S3 guarde los objetos cifrados. Para ello, el administrador de la cuenta A puede usar la clave de condición `s3:x-amz-server-side-encryption`, tal como se muestra. El par clave-valor en el bloque `Condition` especifica la clave `s3:x-amz-server-side-encryption`.

```
"Condition": {
     "StringNotEquals": {
         "s3:x-amz-server-side-encryption": "AES256"
     }}
```

Al probar el permiso con la AWS CLI, debe usar el parámetro requerido utilizando el parámetro `--server-side-encryption`.

```
aws s3api put-object --bucket example1bucket --key HappyFace.jpg --body c:
\HappyFace.jpg --server-side-encryption "AES256" --profile AccountBadmin
```

Ejemplo 2: Concesión de permisos s3:PutObject para copiar objetos con una restricción en el origen de la copia

Se produce una operación de copia cuando usted especifica un objeto de origen en la solicitud PUT Object (consulte PUT Object - Copy). Por lo tanto, el propietario del bucket puede conceder un permiso al usuario para copiar objetos con restricciones en el origen, por ejemplo:

- Permitir copiar objetos solo del bucket `sourcebucket`.

- Permitir copiar objetos del bucket de origen y solo los objetos cuyo prefijo de nombre de clave comience por public/ f (por ejemplo: sourcebucket/public/*).
- Permitir copiar solo un objeto específico del bucket de origen (por ejemplo: sourcebucket/example.jpg).

La siguiente política de bucket concede al usuario Dave el permiso `s3:PutObject`. Le permite copiar solo los objetos con la condición de que la solicitud incluya el encabezado `s3:x-amz-copy-source` y el valor del encabezado especifique el prefijo de nombre de clave `/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/*"
                }
            }
        }
    ]
}
```

Prueba de la política con la AWS CLI

Puede usar el comando `copy-object` de la AWS CLI para probar el permiso. Para especificar el origen, debe añadir el parámetro `--copy-source` y el prefijo de nombre de clave debe coincidir con el prefijo permitido en la política. Tiene que proporcionar las credenciales al usuario Dave utilizando el parámetro `--profile`. Para obtener más información acerca de la configuración de la AWS CLI, consulte Desarrollo con Amazon S3 mediante la AWS CLI.

```
aws s3api copy-object --bucket awsexamplebucket1 --key HappyFace.jpg
--copy-source examplebucket/public/PublicHappyFace1.jpg --profile AccountADave
```

Conceder permiso para copiar solo un objeto específico

La política anterior utiliza la condición `StringNotLike`. Para conceder el permiso para copiar solo un objeto específico, debe cambiar la condición de `StringNotLike` a `StringNotEquals` y, a continuación, especificar la clave exacta del objeto, tal como se muestra.

```
"Condition": {
       "StringNotEquals": {
           "s3:x-amz-copy-source": "awsexamplebucket1/public/PublicHappyFace1.jpg"
       }
}
```

Ejemplo 3: Concesión de acceso a una versión específica de un objeto

Supongamos que la cuenta A tiene un bucket con control de versiones habilitado. El bucket tiene varias versiones del objeto `HappyFace.jpg`. Ahora, el administrador de la cuenta desea conceder al usuario (Dave) permiso para obtener solo una versión específica del objeto. Para ello, el administrador de la cuenta debe conceder a Dave el permiso `s3:GetObjectVersion` de forma condicional, tal como se muestra. El par clave-valor en el bloque `Condition` especifica la clave de condición `s3:VersionId`. En este caso, Dave tiene que saber el ID de versión exacta del objeto para recuperar el objeto.

Para obtener más información, consulte GetObject en la referencia de la API de Amazon Simple Storage Service.

```
{
    "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"
                }
            }
        }
    ]
}
```

Prueba de la política con la AWS CLI

Puede usar el comando `get-object` de la AWS CLI para probar estos permisos y el parámetro `--version-id` para identificar la versión específica del objeto. El comando recupera el objeto y lo guarda en el archivo `OutputFile.jpg`.

```
aws s3api get-object --bucket examplebucketversionenabled --key HappyFace.jpg
 OutputFile.jpg --version-id AaaHbAQitwiL_h47_44lRO2DDfLlBO5e --profile AccountADave
```

Ejemplo 4: Concesión de permisos basados en etiquetas de objeto

Para obtener ejemplos sobre cómo utilizar claves de condición de etiquetado de objetos con operaciones de Amazon S3, consulte Etiquetado y políticas de control de acceso.

Ejemplo 5: Restricción del acceso mediante el ID de Cuenta de AWS del propietario del bucket

Puede utilizar la clave `aws:ResourceAccount` o `s3:ResourceAccount` para escribir políticas de punto de conexión de IAM o nube privada virtual (VPC) que restrinjan el acceso de usuarios, roles o aplicaciones a los buckets de Amazon S3 que pertenecen a un ID de Cuenta de AWS específico.

Puede usar esta clave de condición para restringir que los clientes dentro de su VPC accedan a los buckets que no posee.

Sin embargo, debe tener en cuenta que algunos servicios de AWS dependen del acceso a buckets administrados de AWS. Por lo tanto, es posible que el uso de la clave `aws:ResourceAccount` o `s3:ResourceAccount` de la política de IAM también afecte al acceso a estos recursos.

Para obtener más información y ejemplos, consulte los siguientes recursos:

- Restringir el acceso a los buckets de una Cuenta de AWS específica en la Guía de AWS PrivateLink
- Restringir el acceso a los buckets que utiliza Amazon ECR en la Guía de Amazon ECR
- Proporcionar el acceso necesario a Systems Manager para buckets de Amazon S3 administrados por AWS en la Guía de AWS Systems Manager
- Limitar el acceso a los buckets de Amazon S3 propiedad de Cuentas de AWS específicas en el Blog de almacenamiento de AWS

Ejemplo 6: Necesidad de una versión mínima de TLS

Si utiliza la nueva clave de condición de IAM s3:TLSVersion para escribir IAM, Virtual Private Cloud Endpoint (VPCE) o políticas de bucket que restrinjan el acceso de aplicaciones o usuarios a buckets de Amazon S3 en función de la versión de TLS que utilice el cliente. Puede utilizar esta clave de condición para escribir políticas que requieran una versión mínima de TLS.

Example

Esta política de bucket de ejemplo deniega las solicitudes de PutObject de clientes que tienen una versión TLS inferior a 1.2, por ejemplo, 1.1 o 1.0.

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Effect": "Deny",
            "Principal": "*",
            "Action": "s3:PutObject",
            "Resource": [
                "arn:aws:s3:::amzn-s3-demo-bucket1",
```

```
                "arn:aws:s3:::amzn-s3-demo-bucket1/*"
            ],
            "Condition": {
                "NumericLessThan": {
                    "s3:TlsVersion": 1.2
                }
            }
        }
    ]
}
```

Example

Esta política de bucket de ejemplo permite solicitudes de PutObject de clientes que tienen una versión TLS superior a 1.1, por ejemplo, 1.2, 1.3 o superior.

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Principal": "*",
            "Action": "s3:PutObject",
            "Resource": [
                "arn:aws:s3:::amzn-s3-demo-bucket1",
                "arn:aws:s3:::amzn-s3-demo-bucket1/*"
            ],
            "Condition": {
                "NumericGreaterThan": {
                    "s3:TlsVersion": 1.1
                }
            }
        }
    ]
}
```

Ejemplos: claves de condición de Amazon S3 para operaciones con buckets

Esta sección incluye políticas de ejemplo que muestran cómo puede utilizar claves de condición específicas de Amazon S3 para operaciones con buckets.

Temas

Ejemplo 1: Concesión de permisos s3:GetObject con una condición en una dirección IP

Puede conceder permiso a los usuarios autenticados para usar la acción `s3:GetObject` si la solicitud se genera a partir de un intervalo específico de direcciones IP (192.0.2.*), a menos que la dirección IP sea 192.0.2.188. En el bloque de condición, `IpAddress` y `NotIpAddress` son condiciones y cada una recibe un par clave-valor para evaluación. En este ejemplo, los pares de clave-valor usan la clave general de AWS `aws:SourceIp`.

> **Note**
>
> Los valores de clave `IPAddress` y `NotIpAddress` especificados en la condición utilizan la notación CIDR como se describe en RFC 4632. Para obtener más información, consulte 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"
                }
            }
        }
```

```
    ]
}
```

También puede utilizar otras claves de condición generales de AWS en las políticas de Amazon S3. Por ejemplo, puede especificar las claves de condición `aws:SourceVpce` y `aws:SourceVpc` en las políticas de bucket para los puntos de enlace de la VPC. Para ver ejemplos específicos, consulte Control del acceso desde puntos de enlace de la VPC con políticas de bucket.

> **Note**
>
> Para alguna claves de condición globales de AWS, solo se admiten determinados tipos de recursos. Por lo tanto, compruebe si Amazon S3 admite la clave de condición global y el tipo de recurso que desea usar o si necesitará usar una clave de condición específica de Amazon S3 en su lugar. Para obtener una lista completa de los tipos de recursos y claves de condición admitidos para Amazon S3, consulte Actions, resources, and condition keys for Amazon S3 en la Referencia de autorización de servicios.

Ejemplo 2: obtener una lista de objetos en un bucket con un prefijo específico

Puede utilizar la clave de condición de `s3:prefix` para limitar la respuesta de la API GET Bucket (ListObjects) a nombres de clave con un prefijo específico. Si es el propietario del bucket, puede restringir el permiso para que un usuario pueda enumerar el contenido de un prefijo específico en el bucket. Esta clave de condición resulta útil si los objetos en el bucket están organizados por prefijos de nombre de clave. La consola de Amazon S3 utiliza prefijos de nombre de clave para mostrar un concepto de carpeta. Solo la consola admite el concepto de carpetas; la API de Amazon S3 solo admite buckets y objetos. Para obtener más información sobre el uso de prefijos y delimitadores para filtrar permisos de acceso, consulte Controlar el acceso a un bucket con las políticas de usuario.

Por ejemplo, tiene dos objetos con nombres de clave `public/object1.jpg` y `public/object2.jpg`, la consola muestra los objetos en la carpeta `public`. En la API de Amazon S3, estos son objetos con prefijos, no objetos en carpetas. Sin embargo, en la API de Amazon S3, si organiza las claves de objetos, puede conceder el permiso `s3:ListBucket` con la condición `s3:prefix` que permitirá al usuario obtener una lista de nombres de clave con esos prefijos específicos.

En este ejemplo, la cuenta del propietario del bucket y la cuenta principal, a la que pertenece el usuario, son las mismas. Por lo tanto, el propietario del bucket puede usar una política de bucket o

de usuario. Para obtener más información acerca de otras claves de condición que puede utilizar con la API GET Bucket (ListObjects), consulte ListObjects.

Política de usuario

La siguiente política de usuario concede el permiso `s3:ListBucket` (consulte GET Bucket (List Objects)) con una condición que requiere que el usuario especifique `prefix` en la solicitud con el valor `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 condición impide que el usuario cree una lista de claves de objetos con el prefijo `projects`. Esta denegación explícita añadida impide que el usuario realice una lista de claves con cualquier otro prefijo, independientemente de los demás permisos que tenga el usuario. Por ejemplo, es posible que el usuario obtenga el permiso para crear una lista de claves de objetos sin ninguna restricción mediante actualizaciones de políticas de usuario anteriores o mediante una política de bucket. Sin

embargo, como la denegación explícita siempre sustituye a cualquier otro permiso, la solicitud del usuario para crear listas de claves que no tengan el prefijo `projects` se deniega.

Política de bucket

Si añade el elemento `Principal` a la política de usuario anterior, que identifica al usuario, tendrá una política de bucket, tal como se muestra.

```
{
   "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"
             }
          }
       }
    ]
}
```

Prueba de la política con la AWS CLI

Puede usar el siguiente comando `list-object` de la AWS CLI para probar la política. En el comando, usa el parámetro `--profile` para proporcionar las credenciales de usuario. Para obtener más información acerca de la configuración y el uso de la AWS CLI, consulte Desarrollo con Amazon S3 mediante la AWS CLI.

```
aws s3api list-objects --bucket awsexamplebucket1 --prefix examplefolder --profile
 AccountADave
```

Si el bucket tiene el control de versiones habilitado, para crear una lista de los objetos en el bucket, en vez del permiso `s3:ListBucketVersions`, debe conceder el permiso `s3:ListBucket` en la política anterior. Este permiso también admite la clave de condición `s3:prefix`.

Ejemplo 3: establecer el número máximo de claves

Puede utilizar la clave de condición `s3:max-keys` para establecer el número máximo de claves que el solicitante puede devolver en una solicitud GET Bucket (ListObjects) o ListObjectVersions. De forma predeterminada, la API devuelve hasta 1000 claves. Para obtener una lista de los operadores de condición numéricos que se pueden utilizar con `s3:max-keys` y los ejemplos adjuntos, consulte Operadores de condición numérica en la guía del usuario de IAM.

Políticas basadas en identidad para Amazon S3

De forma predeterminada, los usuarios y roles no tienen permiso para crear ni modificar los recursos de Amazon S3. Tampoco pueden realizar tareas mediante la AWS Management Console, la AWS Command Line Interface (AWS CLI) o la API de AWS. Un administrador de IAM puede crear políticas de IAM para conceder permisos a los usuarios para realizar acciones en los recursos que necesitan. A continuación, el administrador puede añadir las políticas de IAM a roles y los usuarios pueden asumirlos.

Para obtener información acerca de cómo crear una política basada en identidades de IAM mediante el uso de estos documentos de políticas JSON de ejemplo, consulte Creación de políticas de IAM en la Guía del usuario de IAM.

A fin de obtener más información sobre las acciones y los tipos de recursos definidos por Amazon S3, incluido el formato de los ARN para cada tipo de recurso, consulte Acciones, recursos y claves de condición para Amazon S3 en la Referencia de autorizaciones de servicio.

Temas

- Prácticas recomendadas sobre las políticas

- Ejemplos de políticas basadas en identidad para Amazon S3
- Controlar el acceso a un bucket con las políticas de usuario

Prácticas recomendadas sobre las políticas

Las políticas basadas en identidad determinan si alguien puede crear, acceder o eliminar los recursos de Amazon S3 de la cuenta. Estas acciones pueden generar costos adicionales para su Cuenta de AWS. Siga estas directrices y recomendaciones al crear o editar políticas basadas en identidades:

- Comience con las políticas administradas por AWSy continúe con los permisos de privilegio mínimo: a fin de comenzar a conceder permisos a los usuarios y las cargas de trabajo, utilice las políticas administradas por AWS, que conceden permisos para muchos casos de uso comunes. Están disponibles en su Cuenta de AWS. Se recomienda definir políticas administradas por el cliente de AWS específicas para sus casos de uso a fin de reducir aún más los permisos. Con el fin de obtener más información, consulte las políticas administradas por AWS o las políticas administradas por AWS para funciones de trabajo en la Guía de usuario de IAM.
- Aplique permisos de privilegio mínimo: cuando establezca permisos con políticas de IAM, conceda solo los permisos necesarios para realizar una tarea. Para ello, debe definir las acciones que se pueden llevar a cabo en determinados recursos en condiciones específicas, también conocidos como permisos de privilegios mínimos. Con el fin de obtener más información sobre el uso de IAM para aplicar permisos, consulte Políticas y permisos en IAM en la Guía del usuario de IAM.
- Utilice condiciones en las políticas de IAM para restringir aún más el acceso: puede agregar una condición a sus políticas para limitar el acceso a las acciones y los recursos. Por ejemplo, puede escribir una condición de políticas para especificar que todas las solicitudes deben enviarse utilizando SSL. También puede usar condiciones para conceder acceso a acciones de servicios si se emplean a través de un Servicio de AWS determinado como, por ejemplo, AWS CloudFormation. Para obtener más información, consulte Elementos de la política de JSON de IAM: Condición en la Guía del usuario de IAM.
- Utilice el analizador de acceso de IAM para validar las políticas de IAM con el fin de garantizar la seguridad y funcionalidad de los permisos: el analizador de acceso de IAM valida políticas nuevas y existentes para que respeten el lenguaje (JSON) de las políticas de IAM y las prácticas recomendadas de IAM. El analizador de acceso de IAM proporciona más de 100 verificaciones de políticas y recomendaciones procesables para ayudar a crear políticas seguras y funcionales. Para más información, consulte Política de validación de Analizador de acceso de IAM en la Guía de usuario de IAM.

- Solicite la autenticación multifactor (MFA): si se encuentra en una situación en la que necesita usuarios raíz o de IAM en su Cuenta de AWS, active la MFA para mayor seguridad. Para solicitar la MFA cuando se invocan las operaciones de la API, agregue las condiciones de la MFA a sus políticas. Para más información, consulte Configuración del acceso a una API protegido por MFA en la Guía de usuario de IAM.

Para obtener más información sobre las prácticas recomendadas de IAM, consulte las Prácticas recomendadas de seguridad en IAM en la Guía del usuario de IAM.

Ejemplos de políticas basadas en identidad para Amazon S3

En esta sección se muestran varias políticas basadas en identidad de AWS Identity and Access Management (IAM) de ejemplo para controlar el acceso a Amazon S3. Para ver ejemplos de políticas de bucket (políticas basadas en recursos), consulte Políticas de buckets para Amazon S3. Para obtener información sobre el lenguaje de políticas de IAM, consulte Políticas y permisos en Amazon S3.

Las siguientes políticas de ejemplo funcionarán si las prueba mediante programación. Sin embargo, debe conceder permisos adicionales que necesita la consola de Amazon S3 para utilizarlas con dicha consola. Para obtener información acerca del uso de estas políticas con la consola de Amazon S3, consulte Controlar el acceso a un bucket con las políticas de usuario.

Temas

- Permiso para que el usuario de IAM tenga acceso a uno de los buckets
- Permiso para que cada usuario de IAM tenga acceso a una carpeta en un bucket
- Permiso para que un grupo tenga una carpeta compartida en Amazon S3
- Permiso para que los usuarios lean objetos en una parte del bucket
- Permiso para que un socio coloque archivos en una parte específica de un bucket
- Restringir el acceso a los buckets de Amazon S3 en una Cuenta de AWS específica
- Restricción del acceso a buckets de Amazon S3 dentro de la unidad organizativa
- Restricción del acceso a buckets de Amazon S3 dentro de la organización
- Concesión de permiso para recuperar la configuración de PublicAccessBlock de una Cuenta de AWS
- Restricción de la creación de buckets a una región

Permiso para que el usuario de IAM tenga acceso a uno de los buckets

En este ejemplo, desea conceder a un usuario de IAM en el acceso de su Cuenta de AWS a uno de sus buckets, *amzn-s3-demo-bucket1*, y permitirle que agregue, actualice y elimine objetos.

Además de conceder los permisos `s3:PutObject`, `s3:GetObject` y `s3:DeleteObject` al usuario, la política también concede los permisos `s3:ListAllMyBuckets`, `s3:GetBucketLocation` y `s3:ListBucket`. Estos son los permisos adicionales que requiere la consola. Las acciones `s3:PutObjectAcl` y `s3:GetObjectAcl` también son necesarias para poder copiar, cortar y pegar objetos en la consola. Para ver un ejemplo de un tutorial en el que se conceden permisos a usuarios y se los prueba con la consola, consulte Controlar el acceso a un bucket con las políticas de usuario.

```
{
   "Version":"2012-10-17",
   "Statement":[
      {
         "Effect":"Allow",
         "Action": "s3:ListAllMyBuckets",
         "Resource":"*"
      },
      {
         "Effect":"Allow",
         "Action":["s3:ListBucket","s3:GetBucketLocation"],
         "Resource":"arn:aws:s3:::amzn-s3-demo-bucket1"
      },
      {
         "Effect":"Allow",
         "Action":[
            "s3:PutObject",
            "s3:PutObjectAcl",
            "s3:GetObject",
            "s3:GetObjectAcl",
            "s3:DeleteObject"
         ],
         "Resource":"arn:aws:s3:::amzn-s3-demo-bucket1/*"
      }
   ]
}
```

Permiso para que cada usuario de IAM tenga acceso a una carpeta en un bucket

En este ejemplo, desea conceder el acceso al bucket *amzn-s3-demo-bucket1* a dos usuarios de IAM, Mary y Carlos, para que puedan agregar, actualizar y eliminar objetos. Sin embargo, desea impedir que cada usuario acceda a un único prefijo (carpeta) en el bucket. Debe crear carpetas con nombres que coincidan con los nombres de usuarios.

```
amzn-s3-demo-bucket1
   Mary/
   Carlos/
```

Para conceder a cada usuario el acceso solo a sus carpetas, puede escribir una política para cada usuario y asociarla de forma individual. Por ejemplo, puede asociar la siguiente política al usuario Mary para conceder permisos específicos de Amazon S3 en la carpeta *amzn-s3-demo-bucket1*/*Mary*.

```
{
   "Version":"2012-10-17",
   "Statement":[
      {
         "Effect":"Allow",
         "Action":[
            "s3:PutObject",
            "s3:GetObject",
            "s3:GetObjectVersion",
            "s3:DeleteObject",
            "s3:DeleteObjectVersion"
         ],
         "Resource":"arn:aws:s3:::amzn-s3-demo-bucket1/Mary/*"
      }
   ]
}
```

Puede asociar una política similar al usuario Carlos, especificando la carpeta *Carlos* en el valor `Resource`.

En vez de asociar políticas a cada usuario, puede escribir una sola política que utilice una variable de política y, a continuación, asociarla a un grupo. En primer lugar, debe crear un grupo y agregar a Mary y Carlos al grupo. La política del ejemplo siguiente muestra cómo conceder un conjunto de permisos de Amazon S3 en la carpeta *amzn-s3-demo-bucket1*/`${aws:username}`. Cuando

se evalúa la política, la variable de la política `${aws:username}` se sustituye por el nombre de usuario del solicitante. Por ejemplo, si Mary envía una solicitud para colocar un objeto, la operación se permitirá solo si Mary carga el objeto en la carpeta *amzn-s3-demo-bucket1*/Mary.

```
{
   "Version":"2012-10-17",
   "Statement":[
      {
         "Effect":"Allow",
         "Action":[
            "s3:PutObject",
            "s3:GetObject",
            "s3:GetObjectVersion",
            "s3:DeleteObject",
            "s3:DeleteObjectVersion"
         ],
         "Resource":"arn:aws:s3:::amzn-s3-demo-bucket1/${aws:username}/*"
      }
   ]
}
```

> **Note**
>
> Cuando utiliza variables de políticas, debe especificar de forma explícita la versión `2012-10-17` en la política. La versión predeterminada del lenguaje de la política de IAM, 2008-10-17, no admite variables de políticas.

Si desea probar la política anterior en la consola de Amazon S3, esta requiere permiso para permisos adicionales, como se muestra en la siguiente política. Para obtener información acerca de cómo la consola utiliza estos permisos, consulte Controlar el acceso a un bucket con las políticas de usuario.

```
{
 "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:::amzn-s3-demo-bucket1",
      "Condition":{
            "StringEquals":{
                   "s3:prefix":[""], "s3:delimiter":["/"]
                         }
               }
    },
    {
      "Sid": "AllowListBucketOfASpecificUserPrefix",
      "Action": "s3:ListBucket",
      "Effect": "Allow",
      "Resource": "arn:aws:s3:::amzn-s3-demo-bucket1",
      "Condition":{  "StringLike":{"s3:prefix":["${aws:username}/*"] }
       }
    },
      {
     "Sid": "AllowUserSpecificActionsOnlyInTheSpecificUserPrefix",
        "Effect":"Allow",
        "Action":[
           "s3:PutObject",
           "s3:GetObject",
           "s3:GetObjectVersion",
           "s3:DeleteObject",
           "s3:DeleteObjectVersion"
        ],
        "Resource":"arn:aws:s3:::amzn-s3-demo-bucket1/${aws:username}/*"
      }
  ]
}
```

> **Note**
>
> En la versión 2012-10-17 de la política, las variables comienzan con $. Este cambio en la sintaxis puede crear un conflicto potencial si la clave del objeto (nombre de objeto) incluye $.

> Para evitar este conflicto, especifique el carácter $ mediante ${$}. Por ejemplo, para incluir la clave de objeto my$file en la política, debe especificarlo como my${$}file.

Aunque los nombres de usuario de IAM son identificadores descriptivos fáciles, no necesariamente deben ser identificadores globales únicos. Por ejemplo, si el usuario Carlos abandona la organización y se une un usuario con el mismo nombre (Carlos), este puede tener acceso a la información del usuario anterior con el mismo nombre.

En vez de usar nombres de usuario, puede crear carpetas en función de los ID de usuario de IAM. Cada ID de usuario de IAM es único. En este caso, debe modificar la política anterior para usar la variable de política ${aws:userid}. Para obtener más información acerca de los identificadores de usuarios, consulte Identificadores de IAM en la guía de usuario de IAM.

```
{
   "Version":"2012-10-17",
   "Statement":[
      {
         "Effect":"Allow",
         "Action":[
            "s3:PutObject",
            "s3:GetObject",
            "s3:GetObjectVersion",
            "s3:DeleteObject",
            "s3:DeleteObjectVersion"
         ],
         "Resource":"arn:aws:s3:::amzn-s3-demo-bucket1/home/${aws:userid}/*"
      }
   ]
}
```

Permiso para que los usuarios que no son de IAM (usuarios de aplicaciones móviles) tengan acceso a las carpetas de un bucket

Supongamos que desea desarrollar una aplicación móvil, un juego que almacena los datos de usuario en un bucket de S3. Debe crear una carpeta en el bucket para cada usuario de la aplicación. También debe limitar el acceso de cada usuario a su propia carpeta. Pero no puede crear carpetas antes de que alguien descargue la aplicación y comience a jugar porque no tiene su ID de usuario.

En este caso, puede solicitar a los usuarios que inicien sesión en la aplicación con los proveedores de identidades públicos como Login with Amazon, Facebook o Google. Una vez que los usuarios

iniciaron sesión en la aplicación mediante uno de estos proveedores, tendrán un ID de usuario que usted podrá utilizar para crear carpetas específicas de usuarios en tiempo de ejecución.

Luego, puede utilizar la identidad federada de sitio web en AWS Security Token Service para integrar la información del proveedor de identidad con su aplicación y obtener credenciales de seguridad temporales para cada usuario. A continuación, puede crear políticas de IAM para permitir que la aplicación tenga acceso a su bucket y realice operaciones como crear carpetas específicas de usuario y cargar datos. Para obtener más información acerca de la federación de identidades de sitio web, consulte Acerca de federación de identidades de sitio web en la Guía del usuario de IAM.

Permiso para que un grupo tenga una carpeta compartida en Amazon S3

Al asociar la siguiente política a un grupo, todos los miembros del grupo obtienen acceso a la siguiente carpeta en Amazon S3: *amzn-s3-demo-bucket1*/share/marketing. Los miembros del grupo solo pueden acceder a los permisos específicos de Amazon S3 que se muestran en la política y solo a los objetos en la carpeta especificada.

```
{
   "Version":"2012-10-17",
   "Statement":[
      {
         "Effect":"Allow",
         "Action":[
            "s3:PutObject",
            "s3:GetObject",
            "s3:GetObjectVersion",
            "s3:DeleteObject",
            "s3:DeleteObjectVersion"
         ],
         "Resource":"arn:aws:s3:::amzn-s3-demo-bucket1/share/marketing/*"
      }
   ]
}
```

Permiso para que los usuarios lean objetos en una parte del bucket

En este ejemplo, crea un grupo denominado *AllUsers*, que contiene todos los usuarios de IAM que pertenecen a la Cuenta de AWS. A continuación asocia una política que concede al grupo el acceso a GetObject y GetObjectVersion, pero solo para objetos en la carpeta *amzn-s3-demo-bucket1*/readonly.

```
{
   "Version":"2012-10-17",
   "Statement":[
      {
         "Effect":"Allow",
         "Action":[
            "s3:GetObject",
            "s3:GetObjectVersion"
         ],
         "Resource":"arn:aws:s3:::amzn-s3-demo-bucket1/readonly/*"
      }
   ]
}
```

Permiso para que un socio coloque archivos en una parte específica de un bucket

En este ejemplo, crea un grupo llamado *AnyCompany* que representa a una compañía asociada. Crea un usuario de IAM para la persona o aplicación específica en la empresa asociada que necesita acceso y, luego, coloca al usuario en el grupo.

Luego, asocia una política que concede al grupo `PutObject` acceso a la siguiente carpeta en un bucket:

amzn-s3-demo-bucket1`/uploads/anycompany`

Como quiere evitar que el grupo *AnyCompany* realice cualquier otro tipo de acción con el bucket, agrega una instrucción que deniegue explícitamente el permiso para cualquier acción de Amazon S3, excepto `PutObject` en todos los recursos de Amazon S3 en la Cuenta de AWS.

```
{
   "Version":"2012-10-17",
   "Statement":[
      {
         "Effect":"Allow",
         "Action":"s3:PutObject",
         "Resource":"arn:aws:s3:::amzn-s3-demo-bucket1/uploads/anycompany/*"
      },
      {
         "Effect":"Deny",
         "Action":"s3:*",
         "NotResource":"arn:aws:s3:::amzn-s3-demo-bucket1/uploads/anycompany/*"
      }
```

```
    ]
}
```

Restringir el acceso a los buckets de Amazon S3 en una Cuenta de AWS específica

Si desea asegurarse de que las entidades principales de Amazon S3 accedan solo a los recursos que se encuentran dentro de una Cuenta de AWS de confianza, puede restringir el acceso. Por ejemplo, esta Política de IAM basada en identidades utiliza un efecto `Deny` para bloquear el acceso a las acciones de Amazon S3, a menos que el recurso de Amazon S3 al que se accede esté en una cuenta *222222222222*. Para evitar que una entidad principal de IAM en una Cuenta de AWS acceda a objetos de Amazon S3 fuera de la cuenta, asocie la siguiente política de IAM:

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Sid": "DenyS3AccessOutsideMyBoundary",
      "Effect": "Deny",
      "Action": [
        "s3:*"
      ],
      "Resource": "*",
      "Condition": {
        "StringNotEquals": {
          "aws:ResourceAccount": [
            "222222222222"
          ]
        }
      }
    }
  ]
}
```

> **Note**
>
> Esta política no sustituye los controles de acceso de IAM existentes porque no otorga acceso alguno. En cambio, esta política actúa como una barrera de protección adicional para los demás permisos de IAM, independientemente de los permisos otorgados a través de otras políticas de IAM.

Asegúrese de sustituir el ID de cuenta *222222222222* en la política por su propia Cuenta de AWS. Para aplicar una política a varias cuentas sin dejar de mantener esta restricción, sustituya el ID de cuenta por la clave de condición `aws:PrincipalAccount`. Esta condición requiere que la entidad principal y el recurso se encuentren en la misma cuenta.

Restricción del acceso a buckets de Amazon S3 dentro de la unidad organizativa

Si tiene una Unidad organizativa (OU) configure en AWS Organizations, es posible que desee restringir el acceso al bucket de Amazon S3 a una parte específica de la organización. En este ejemplo, utilizaremos la clave `aws:ResourceOrgPaths` para restringir el acceso del bucket de Amazon S3 a una unidad organizativa de la organización. En este ejemplo, el ID de la unidad organizativa es *ou-acroot-exampleou*. Asegúrese de sustituir este valor en su propia política por sus propios ID de unidad organizativa.

```
{
   "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**
>
> Esta política no concede ningún acceso. En su lugar, esta política actúa como respaldo para los demás permisos de IAM, lo que impide que las entidades principales accedan a objetos de Amazon S3 fuera de un límite definido por la unidad organizativa.

La política niega el acceso a las acciones de Amazon S3 a menos que el objeto de Amazon S3 al que se accede se encuentre en la unidad organizativa *ou-acroot-exampleou* de la organización. La condición de política de IAM requiere `aws:ResourceOrgPaths`, una clave de condición multivalor, para contener cualquiera de las rutas de unidad organizativa mostradas. La política utiliza el operador `ForAllValues:StringNotLike` para comparar los valores de `aws:ResourceOrgPaths` con las unidades organizativas mostradas sin la coincidencia que distingue entre mayúsculas y minúsculas.

Restricción del acceso a buckets de Amazon S3 dentro de la organización

Para restringir el acceso a los objetos de Amazon S3 dentro de la organización, asocie una política de IAM a la raíz de la organización, aplicándola a todas las cuentas de la organización. Para requerir a las entidades principales de IAM que sigan esta regla, utilice una política de control de servicio (SCP). Si elige utilizar SCP, asegúrese de probar SCP minuciosamente antes de adjuntar la política a la raíz de la organización.

En la siguiente política de ejemplo, se deniega el acceso a las acciones de Amazon S3 a menos que el objeto de Amazon S3 al que se accede esté en la misma organización que la entidad principal de IAM que accede a él:

```
{
   "Version": "2012-10-17",
   "Statement": [
     {
       "Sid": "DenyS3AccessOutsideMyBoundary",
       "Effect": "Deny",
       "Action": [
         "s3:*"
       ],
       "Resource": "arn:aws:s3:::*/*",
       "Condition": {
         "StringNotEquals": {
           "aws:ResourceOrgID": "${aws:PrincipalOrgID}"
         }
       }
     }
   ]
 }
```

> **Note**
>
> Esta política no concede ningún acceso. En su lugar, esta política actúa como respaldo para los demás permisos de IAM, lo que impide que las entidades principales accedan a objetos de Amazon S3 fuera de la organización. Esta política también se aplica a los recursos de Amazon S3 que se crean después de que la política haya entrado en vigor.

La Condición de política de IAM en este ejemplo requiere `aws:ResourceOrgID` y `aws:PrincipalOrgID` para ser iguales entre sí. Con este requisito, la entidad principal que realiza la solicitud y el recurso al que se accede deben estar en la misma organización.

Concesión de permiso para recuperar la configuración de PublicAccessBlock de una Cuenta de AWS

La política basada en identidad de ejemplo siguiente concede el permiso `s3:GetAccountPublicAccessBlock` a un usuario. Para todos estos permisos, estableció el valor `Resource` en `"*"`. Para obtener información sobre los ARN de recursos, consulte Recursos de políticas para Amazon S3.

```
{
   "Version":"2012-10-17",
   "Statement":[
      {
         "Sid":"statement1",
         "Effect":"Allow",
         "Action":[
            "s3:GetAccountPublicAccessBlock"
         ],
         "Resource":[
            "*"
         ]
       }
    ]
}
```

Restricción de la creación de buckets a una región

Supongamos que el administrador de una Cuenta de AWS desea conceder al usuario (Dave) permiso para crear un bucket solo en la región de América del Sur (São Paulo). El administrador de la cuenta puede asociar la siguiente política de usuario que concede el permiso `s3:CreateBucket`

con una condición, tal como se muestra. El par clave-valor en el bloque `Condition` especifica la clave `s3:LocationConstraint` y la región `sa-east-1` como su valor.

> **Note**
>
> En este ejemplo, el propietario del bucket concede el permiso a uno de sus usuarios, así que se puede usar una política de bucket o de usuario. En este ejemplo se muestra una política de usuario.

Para obtener una lista de las regiones de Amazon S3, consulte Regiones y puntos de conexión en la Referencia general de AWS.

```
{
   "Version":"2012-10-17",
   "Statement":[
      {
         "Sid":"statement1",
         "Effect":"Allow",
         "Action": "s3:CreateBucket",
         "Resource": "arn:aws:s3:::*",
         "Condition": {
             "StringLike": {
                 "s3:LocationConstraint": "sa-east-1"
             }
         }
       }
    ]
}
```

Agregar denegación de forma explícita

La política anterior impide que el usuario cree un bucket en cualquier otra región, excepto la región `sa-east-1`. Sin embargo, es posible que otras políticas concedan al usuario el permiso para crear buckets en otra región. Por ejemplo, si el usuario pertenece a un grupo, es posible que el grupo tenga una política asociada que permite a todos los usuarios en el grupo crear buckets en otra región. Para asegurarse de que el usuario no obtenga el permiso para crear buckets en otra región, puede agregar una instrucción de denegación explícita en la política anterior.

La instrucción `Deny` utiliza la condición `StringNotLike`. Es decir, la solicitud para crear un bucket se deniega si la restricción de ubicación no es `sa-east-1`. La denegación explícita no permite al

usuario crear un bucket en ninguna otra región, independientemente de los permisos que obtenga el usuario. La siguiente política incluye una instrucción de denegación explícita.

```
{
   "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"
             }
         }
       }
    ]
}
```

Prueba de la política con la AWS CLI

Puede usar el siguiente comando `create-bucket` de la AWS CLI para probar la política. En este ejemplo se utiliza el archivo `bucketconfig.txt` para especificar la restricción de ubicación. Tenga en cuenta la ruta del archivo de Windows. Tiene que actualizar el nombre y la ruta del bucket según corresponda. Debe usar el parámetro `--profile` para proporcionar las credenciales de usuario. Para obtener más información acerca de la configuración y el uso de la AWS CLI, consulte Desarrollo con Amazon S3 mediante la AWS CLI.

```
aws s3api create-bucket --bucket examplebucket --profile AccountADave --create-bucket-
configuration file://c:/Users/someUser/bucketconfig.txt
```

El archivo `bucketconfig.txt` especifica la configuración, tal y como se muestra a continuación.

```
{"LocationConstraint": "sa-east-1"}
```

Controlar el acceso a un bucket con las políticas de usuario

En este tutorial se explica cómo funcionan los permisos del usuario con Amazon S3. En este ejemplo, puede crear un bucket con carpetas. A continuación, cree usuarios de IAM AWS Identity and Access Management en la Cuenta de AWS y conceda a estos usuarios permisos progresivos en el bucket de Amazon S3 y sus carpetas.

Temas

Conceptos básicos de buckets y carpetas

El modelo de datos de Amazon S3 es una estructura plana: usted crea un bucket y el bucket almacena objetos. No existe una jerarquía entre los subbuckets o las subcarpetas, pero puede emular una jerarquía de carpetas. Las herramientas, como, por ejemplo, la consola de Amazon S3, pueden mostrar una vista de estas carpetas y subcarpetas lógicas del bucket.

La consola muestra que un bucket denominado `companybucket` tiene tres carpetas: `Private`, `Development` y `Finance`, y un objeto, `s3-dg.pdf`. La consola utiliza los nombres de objeto (claves) para crear una jerarquía lógica con carpetas y subcarpetas. Considere los siguientes ejemplos:

- Al crear la carpeta `Development`, la consola crea un objeto con la clave `Development/`. Tenga en cuenta el delimitador final (`/`).
- Al cargar un objeto denominado `Projects1.xls` en la carpeta `Development`, la consola carga el objeto y le asigna la clave `Development/Projects1.xls`.

  En la clave, `Development` es el prefijo y `/` es el delimitador. La API de Amazon S3 admite prefijos y delimitadores en sus operaciones. Por ejemplo, puede obtener una lista de todos los objetos de un bucket con un prefijo y delimitador específicos. En la consola, al abrir la carpeta `Development`, la consola muestra los objetos de esa carpeta. En el siguiente ejemplo, la carpeta `Development` contiene un objeto.

  Cuando la consola muestra la carpeta `Development` del bucket `companybucket`, envía una solicitud a Amazon S3 en la que especifica el prefijo `Development` y el delimitador `/`. La respuesta de la consola se parece a una lista de carpetas del sistema de archivos de su equipo. En el ejemplo anterior se muestra que el bucket `companybucket` tiene un objeto con la clave `Development/Projects1.xls`.

La consola utiliza claves de objeto para inferir una jerarquía lógica. Amazon S3 no tiene una jerarquía física. Amazon S3 solo tiene buckets que contienen objetos en una estructura de archivos plana. Cuando crea objetos con la API de Amazon S3, puede utilizar las claves del objeto que implican una jerarquía lógica. Al crear una jerarquía lógica de objetos, puede administrar el acceso a carpetas individuales, tal y como se explica en este tutorial.

Antes de comenzar, asegúrese de estar familiarizado con el concepto del contenido del bucket en el nivel raíz. Supongamos que el bucket `companybucket` tiene los siguientes objetos:

- `Private/privDoc1.txt`
- `Private/privDoc2.zip`
- `Development/project1.xls`
- `Development/project2.xls`
- `Finance/Tax2011/document1.pdf`
- `Finance/Tax2011/document2.pdf`

- `s3-dg.pdf`

Estas claves de objeto crean una jerarquía lógica con `Private`, `Development` y `Finance` como carpetas en el nivel raíz y `s3-dg.pdf` como un objeto en el nivel raíz. Cuando elige el nombre del bucket en la consola de Amazon S3, los elementos en el nivel raíz aparecen. La consola muestra los prefijos de nivel principal (`Private/`, `Development/` y `Finance/`) como carpetas en el nivel raíz. La clave del objeto `s3-dg.pdf` no tiene prefijo y, por lo tanto, aparece como un elemento en el nivel raíz.

Resumen del tutorial

En este tutorial, creará un bucket con tres carpetas: (`Private`, `Development` y `Finance`).

Tiene dos usuarios, Alice y Bob. Quiere que Alice solo tenga acceso a la carpeta `Development` y que Bob solo tenga acceso a la carpeta `Finance`. Desea mantener privado el contenido de la carpeta `Private`. En el tutorial, para administrar el acceso, cree los usuarios de IAM (el ejemplo utiliza los nombres de usuario Alice y Bob) y conceda los permisos necesarios.

IAM también permite crear grupos de usuarios y conceder permisos a nivel grupal para que se apliquen a todos los usuarios del grupo. Esto lo ayuda a administrar mejor los permisos. Para este ejercicio, Alice y Bob deben tener algunos permisos en común. Por lo tanto, también debe crear un grupo denominado `Consultants` y, a continuación, añadir a Alice y Bob al grupo. En primer lugar, para conceder los permisos, asocie una política de grupo al grupo. A continuación, para añadir los permisos específicos del usuario, asocie políticas a usuarios específicos.

> **Note**
>
> En el tutorial, se utiliza `companybucket` como el nombre del bucket, Alice y Bob como los usuarios de IAM y `Consultants` como el nombre del grupo. Dado que Amazon S3 requiere que los nombres de los buckets sean exclusivos a nivel global, debe crear un nombre de bucket para reemplazar el existente.

Prepararse para el tutorial

En este ejemplo, utilizará las credenciales de su Cuenta de AWS para crear usuarios de IAM. Al principio, estos usuarios no tienen permisos. Conceda permisos a estos usuarios de forma gradual para realizar acciones específicas de Amazon S3. Para probar estos permisos, inicie sesión en

la consola con las credenciales de cada usuario. A medida que concede los permisos de forma progresiva como propietario de la Cuenta de AWS y prueba los permisos como usuario de IAM, tendrá que iniciar y cerrar sesión con diferentes credenciales en cada ocasión. Puede realizar estas pruebas con un navegador, pero el proceso se agilizará si utiliza dos navegadores distintos. Utilice un navegador para conectarse a la AWS Management Console con las credenciales de la Cuenta de AWS y otro navegador para conectarse con las credenciales del usuario de IAM.

Para iniciar sesión en la AWS Management Console con las credenciales de la Cuenta de AWS, vaya a https://console.aws.amazon.com/. Un usuario de IAM no puede iniciar sesión con el mismo enlace. Un usuario de IAM debe utilizar una página de inicio de sesión activada para IAM. Como propietario de la cuenta, puede proporcionar este enlace a los usuarios.

Para obtener más información acerca de IAM, consulte Inicio de sesión en la AWS Management Console en la Guía del usuario de IAM.

Para proporcionar un enlace de inicio de sesión para usuarios de IAM

1. Inicie sesión en la AWS Management Console y abra la consola de IAM en https://console.aws.amazon.com/iam/.
2. En el panel de Navigation (Navegación), elija IAM Dashboard (Panel de IAM).
3. Anote la URL en IAM users sign in link: (Enlace de inicio de sesión de usuarios de IAM:). Proporcionará este enlace a los usuarios de IAM para que inicien sesión en la consola con el nombre de usuario y contraseña de IAM.

Paso 1: Crear un bucket

En este paso, inicie sesión en la consola de Amazon S3 con las credenciales de la Cuenta de AWS, cree un bucket, agregue las carpetas al bucket y cargue uno o dos documentos de ejemplo en cada carpeta.

1. Inicie sesión AWS Management Console Management Console y abra la consola de Amazon S3 en https://console.aws.amazon.com/s3/.
2. Cree un bucket.

   Para obtener instrucciones paso a paso, consulte Crear un bucket.
3. Cargue un documento en el bucket.

   Para este ejercicio, supongamos que tiene el documento `s3-dg.pdf` en el nivel raíz de este bucket. Si carga un documento diferente, sustituya el nombre de archivo por `s3-dg.pdf`.

4. Añada las tres carpetas denominadas `Private`, `Finance` y `Development` al bucket.

   Para obtener instrucciones paso a paso para crear una carpeta, consulte Organización de objetos en la consola de Amazon S3 con carpetas> en la Guía del usuario de la consola de Amazon Simple Storage Service.

5. Cargue uno o dos documentos en cada carpeta.

   Para este ejercicio, supongamos que tiene cargado un par de documentos en cada carpeta, lo que hace que el bucket tenga objetos con las siguientes claves:

   - `Private/privDoc1.txt`
   - `Private/privDoc2.zip`
   - `Development/project1.xls`
   - `Development/project2.xls`
   - `Finance/Tax2011/document1.pdf`
   - `Finance/Tax2011/document2.pdf`
   - `s3-dg.pdf`

   Para obtener instrucciones paso a paso, consulte Carga de objetos.

Paso 2: Crear usuarios y un grupo de IAM

Ahora, utilice la consola de IAM para agregar dos usuarios de IAM, Alice y Bob, a su Cuenta de AWS. Para obtener instrucciones paso a paso, consulte Creación de un usuario de IAM en su Cuenta de AWS en la Guía del usuario de IAM.

Cree también un grupo administrativo denominado `Consultants`. A continuación, agregue los usuarios al grupo. Para obtener instrucciones paso a paso, consulte Creación de un grupo de usuarios de IAM.

> **Warning**
>
> Cuando añade los usuarios y un grupo, no debe asociar ninguna política que conceda permisos a estos usuarios. Al principio, estos usuarios no tienen ningún permiso. En las siguientes secciones se explica cómo conceder los permisos de forma gradual. Primero debe asegurarse de haber asignado contraseñas a estos usuarios de IAM. Utilice estas

credenciales de usuario para probar las acciones de Amazon S3 y comprobar que los permisos funcionen de la forma esperada.

Para obtener instrucciones paso a paso sobre cómo crear un nuevo usuario de IAM, consulte Creación de un usuario de IAM en la Cuenta de AWS en la Guía del usuario de IAM. Cuando cree los usuarios para esta explicación, seleccione Acceso a la AWS Management Console y desmarque Acceso programático.

Para obtener instrucciones paso a paso acerca de cómo crear un grupo administrativo, consulte Creación del primer grupo y usuario administrador de IAM en la guía del usuario de IAM.

Paso 3: Comprobar que los usuarios de IAM no tengan permisos

Si utiliza dos navegadores, ahora puede utilizar el segundo navegador para iniciar sesión en la consola con una de las credenciales de usuario de IAM.

1. Con el enlace de inicio de sesión del usuario de IAM (consulte Para proporcionar un enlace de inicio de sesión para usuarios de IAM), inicie sesión en la AWS Management Console con cualquiera de las credenciales de usuario de IAM.
2. Abra la consola de Amazon S3 en https://console.aws.amazon.com/s3.

   Verifique el mensaje de la consola en el que se que indica que el acceso está denegado.

Ahora, puede comenzar a conceder permisos de forma gradual a los usuarios. En primer lugar, asocie una política de grupo que conceda los permisos que ambos usuarios deben tener.

Paso 4: Conceder permisos en el nivel de grupo

Quiere que los usuarios puedan realizar las siguientes acciones:

- Mostrar todos los buckets que pertenecen a la cuenta principal. Para ello, Bob y Alice deben tener permiso para ejecutar la acción `s3:ListAllMyBuckets`.
- Mostrar los elementos, las carpetas y los objetos del bucket `companybucket` en el nivel raíz. Para ello, Bob y Alice deben tener permiso para ejecutar la acción `s3:ListBucket` en el bucket `companybucket`.

En primer lugar, cree una política que conceda estos permisos y, a continuación, asóciela al grupo `Consultants`.

Paso 4.1: Conceder permiso para mostrar todos los buckets

En este paso, creará una política administrada que conceda a los usuarios los permisos mínimos para que puedan mostrar todos los buckets que pertenecen a la cuenta principal. A continuación, asociará la política al grupo `Consultants`. Cuando adjunta la política administrada a un usuario o grupo, permite al usuario o grupo obtener una lista de buckets que pertenecen a la Cuenta de AWS principal.

1. Inicie sesión en la AWS Management Console y abra la consola de IAM en https://console.aws.amazon.com/iam/.

   > **Note**
   >
   > Dado que concederá permisos al usuario, inicie sesión con las credenciales de su Cuenta de AWS, no como usuario de IAM.

2. Cree la política administrada.

   a. En el panel de navegación de la izquierda, elija Policies (Políticas) y, a continuación, seleccione Create Policy (Crear política).

   b. Seleccione la pestaña JSON.

   c. Copie la siguiente política de acceso y péguela en el campo de texto de la política.

   ```
   {
     "Version": "2012-10-17",
     "Statement": [
       {
         "Sid": "AllowGroupToSeeBucketListInTheConsole",
         "Action": ["s3:ListAllMyBuckets"],
         "Effect": "Allow",
         "Resource": ["arn:aws:s3:::*"]
       }
     ]
   }
   ```

   Una política es un documento JSON. En el documento, `Statement` es una matriz de objetos, cada uno describe un permiso con una recopilación de pares nombre-valor. La

política anterior describe un permiso específico. El elemento `Action` especifica el tipo de acceso. En la política, `s3:ListAllMyBuckets` es una acción predeterminada de Amazon S3. Esta acción abarca la operación GET Service de Amazon S3, que devuelve una lista de todos los buckets que pertenecen al remitente autenticado. El valor del elemento `Effect` determina si se concede o deniega un permiso específico.

d. Elija Review Policy (Revisar la política). En la página siguiente, introduzca `AllowGroupToSeeBucketListInTheConsole` en el campo Name (Nombre) y, a continuación, seleccione Create policy (Crear política).

> **Note**
>
> La entrada Summary (Resumen) muestra un mensaje que indica que la política no concede ningún permiso. Para este tutorial, puede hacer caso omiso de este mensaje.

3. Asocie la política administrada `AllowGroupToSeeBucketListInTheConsole` que creó para el grupo `Consultants`.

   Para obtener instrucciones paso a paso acerca de cómo asociar una política administrada, consulte Agregación y eliminación de permisos de identidad de IAM en la Guía del usuario de IAM.

   Asocie los documentos de la política a los usuarios y grupos de IAM en la consola de IAM. Asocie la política al grupo porque quiere que ambos usuarios puedan mostrar los buckets.

4. Pruebe el permiso.

   a. Con el enlace de inicio de sesión del usuario de IAM (consulte Para proporcionar un enlace de inicio de sesión para usuarios de IAM), inicie sesión en la consola con cualquiera de las credenciales del usuario de IAM.

   b. Abra la consola de Amazon S3 en https://console.aws.amazon.com/s3.

      Ahora, la consola debería mostrar todos los buckets pero no los objetos que ellos contienen.

Paso 4.2: Permitir que los usuarios puedan mostrar el contenido en el nivel raíz de un bucket

A continuación, permita a todos los usuarios del grupo `Consultants` mostrar los elementos del bucket `companybucket` en el nivel raíz. Cuando un usuario elija el bucket de la empresa en la consola de Amazon S3, podrá ver los elementos del bucket en el nivel raíz.

> **Note**
>
> En este ejemplo se utiliza `companybucket` a título ilustrativo. Debe utilizar el nombre del bucket que creó.

Para comprender qué solicitud envía la consola a Amazon S3 cuando elige el nombre de un bucket, la respuesta que devuelve Amazon S3 y la forma en que la consola interpreta la respuesta, examine el flujo de forma más detenida.

Al elegir el nombre de un bucket, la consola envía la solicitud GET Bucket (List Objects) a Amazon S3. Esta solicitud incluye los siguientes parámetros:

- El parámetro `prefix` con una cadena vacía como valor.
- El parámetro `delimiter` con `/` como valor.

A continuación, se muestra un ejemplo de solicitud.

```
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 devuelve una respuesta que incluye el siguiente elemento `<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>
```

La clave del objeto `s3-dg.pdf` no contiene el delimitador `/` y Amazon S3 devuelve la clave en el elemento `<Contents>`. Sin embargo, todas las demás claves de nuestro bucket de ejemplo contienen el delimitador `/`. Amazon S3 agrupa estas claves y devuelve un elemento `<CommonPrefixes>` para cada uno de los valores de prefijo diferentes `Development/`, `Finance/` y `Private/`, que es una subcadena desde el comienzo de estas claves hasta la primera instancia del delimitador `/` especificado.

La consola interpreta este resultado y muestra los elementos en el nivel raíz como tres carpetas y una clave de objeto.

Si Bob o Alice abren la carpeta Development (Desarrollo), la consola envía la solicitud GET Bucket (List Objects) a Amazon S3 con los parámetros `prefix` y `delimiter` establecidos en los siguientes valores:

- El parámetro `prefix` con el valor `Development/`.
- El parámetro `delimiter` con el valor "`/`".

Como respuesta, Amazon S3 devuelve las claves de objeto que comienzan con el prefijo especificado.

```
<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 consola muestra las claves de objeto.

Ahora, vuelva a conceder permiso a los usuarios para mostrar los elementos del bucket en el nivel raíz. Para mostrar el contenido del bucket, los usuarios necesitan permiso para ejecutar la acción `s3:ListBucket`, tal como se muestra en la siguiente instrucción de política. Para asegurarse de que vean solo el contenido en el nivel raíz, añada una condición en la que los usuarios deben especificar un parámetro `prefix` vacío en la solicitud; es decir, no pueden hacer doble clic en ninguna de las carpetas en el nivel raíz. Por último, añada una condición para solicitar acceso de tipo carpeta al pedir que las solicitudes de usuario incluyan el parámetro `delimiter` con el valor "/".

```
{
  "Sid": "AllowRootLevelListingOfCompanyBucket",
  "Action": ["s3:ListBucket"],
  "Effect": "Allow",
  "Resource": ["arn:aws:s3:::companybucket"],
  "Condition":{
         "StringEquals":{
             "s3:prefix":[""], "s3:delimiter":["/"]
                       }
              }
}
```

Al elegir un bucket de la consola de Amazon S3, la consola envía primero la solicitud GET Bucket location para buscar la Región de AWS donde se encuentra el bucket. A continuación, la consola utiliza el punto de conexión específico de la región del bucket para enviar la solicitud GET Bucket (List Objects). Como resultado, si los usuarios utilizan la consola, debe concederles permiso para ejecutar la acción `s3:GetBucketLocation`, tal como se muestra en la siguiente instrucción de política.

```
{
   "Sid": "RequiredByS3Console",
   "Action": ["s3:GetBucketLocation"],
   "Effect": "Allow",
   "Resource": ["arn:aws:s3:::*"]
}
```

Para permitir a los usuarios mostrar el contenido del bucket en el nivel raíz

1. Inicie sesión en la AWS Management Console y abra la consola de IAM en https://console.aws.amazon.com/iam/.

Utilice las credenciales de su Cuenta de AWS, no las de usuario de IAM, para iniciar sesión en la consola.

2. Sustituya la política administrada existente `AllowGroupToSeeBucketListInTheConsole` asociada al grupo `Consultants` por la siguiente política, que también permite la acción `s3:ListBucket`. Recuerde sustituir *`companybucket`* en la política `Resource` por el nombre de su bucket.

   Para obtener instrucciones paso a paso, consulte Edición de políticas de IAM en la Guía de usuario de IAM. Al seguir las instrucciones paso a paso, asegúrese de seguir las indicaciones para aplicar los cambios a todas las entidades principales a las que la política está asociada.

```
{
  "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. Pruebe los permisos actualizados.

   a. Con el enlace de inicio de sesión del usuario de IAM (consulte Para proporcionar un enlace de inicio de sesión para usuarios de IAM), inicie sesión en la AWS Management Console.

      Abra la consola de Amazon S3 en https://console.aws.amazon.com/s3.

b. Elija el bucket que creó y la consola mostrará los elementos del bucket en el nivel raíz. Si elige cualquiera de las carpetas del bucket, no podrá ver el contenido de la carpeta, ya que aún no ha concedido esos permisos.

Esta prueba se realiza correctamente cuando los usuarios utilizan la consola de Amazon S3. Cuando elija un bucket en la consola, la implementación de la consola envía una solicitud que incluye el parámetro `prefix`con una cadena vacía como valor y el parámetro `delimiter` con "`/`" como valor.

Paso 4.3: Resumen de la política de grupo

El resultado final de la política de grupo que añadió es conceder a los usuarios de IAM, Alice y Bob, los siguientes permisos mínimos:

- Mostrar todos los buckets que pertenecen a la cuenta principal.
- Ver los elementos del bucket `companybucket` en el nivel raíz.

Sin embargo, los usuarios aún no pueden hacer demasiado. A continuación, conceda permisos específicos del usuario de la siguiente manera:

- Permita a Alice obtener y colocar objetos en la carpeta `Development`.
- Permita a Bob obtener y colocar objetos en la carpeta `Finance`.

Para permisos específicos del usuario, asocie una política al usuario específico, no al grupo. En la siguiente sección, conceda permiso a Alice para trabajar en la carpeta `Development`. Puede repetir los pasos para conceder un permiso similar a Bob para trabajar en la carpeta `Finance`.

Paso 5: Conceder permisos específicos al usuario Alice de IAM

Ahora debe conceder permisos adicionales a Alice para que pueda ver el contenido de la carpeta `Development` y obtener y colocar objetos en esa carpeta.

Paso 5.1: Conceder permiso al usuario Alice de IAM para mostrar el contenido de la carpeta Development (Desarrollo)

Para que Alice pueda mostrar el contenido de la carpeta `Development`, debe aplicar una política al usuario Alice que le conceda permiso para ejecutar la acción `s3:ListBucket` en el bucket `companybucket`, siempre que la solicitud incluya el prefijo `Development/`. Utilice una política insertada, ya que quiere que esta política se aplique únicamente al usuario Alice. Para obtener más

información sobre las políticas insertadas, consulte Políticas administradas e insertadas en la Guía del usuario de IAM.

1. Inicie sesión en la AWS Management Console y abra la consola de IAM en https://console.aws.amazon.com/iam/.

   Utilice las credenciales de su Cuenta de AWS, no las de usuario de IAM, para iniciar sesión en la consola.

2. Cree una política insertada para conceder permiso al usuario Alice para mostrar el contenido de la carpeta `Development`.

   a. En el panel de navegación de la izquierda, elija Users (Usuarios).

   b. Elija el nombre de usuario Alice.

   c. En la página de detalles del usuario, elija la pestaña Permissions (Permisos) y, a continuación, seleccione Add inline policy (Añadir política insertada).

   d. Seleccione la pestaña JSON.

   e. Copie la siguiente política y péguela en el campo de texto de la política.

```
{
    "Version": "2012-10-17",
    "Statement": [
    {
      "Sid": "AllowListBucketIfSpecificPrefixIsIncludedInRequest",
      "Action": ["s3:ListBucket"],
      "Effect": "Allow",
      "Resource": ["arn:aws:s3:::companybucket"],
      "Condition":{  "StringLike":{"s3:prefix":["Development/*"] }
       }
    }
  ]
}
```

   f. Elija Review Policy (Revisar la política). En la página siguiente, introduzca un nombre en el campo Name (Nombre) y, a continuación, elija Create policy (Crear política).

3. Pruebe el cambio en los permisos de Alice:

   a. Con el enlace de inicio de sesión del usuario de IAM (consulte Para proporcionar un enlace de inicio de sesión para usuarios de IAM), inicie sesión en la AWS Management Console.

   b. Abra la consola de Amazon S3 en https://console.aws.amazon.com/s3.

c. En la consola de Amazon S3, compruebe que Alice pueda ver la lista de objetos en la carpeta `Development/` del bucket.

   Cuando el usuario selecciona la carpeta `/Development` para ver la lista de objetos que contiene, la consola de Amazon S3 envía la solicitud `ListObjects` a Amazon S3 con el prefijo `/Development`. Debido a que se le ha concedido permiso al usuario para ver la lista de objetos con el prefijo `Development` y el delimitador `/`, Amazon S3 devuelve la lista de objetos con el prefijo de clave `Development/` y la consola muestra la lista.

Paso 5.2: Conceder permisos al usuario Alice de IAM para obtener y colocar objetos en la carpeta Development (Desarrollo)

Para que Alice pueda obtener y colocar objetos en la carpeta `Development`, necesitará permiso para ejecutar las acciones `s3:GetObject` y `s3:PutObject`. Las siguientes instrucciones de política conceden estos permisos, siempre que la solicitud incluya el parámetro `prefix` con un valor de `Development/`.

```
{
    "Sid":"AllowUserToReadWriteObjectData",
    "Action":["s3:GetObject", "s3:PutObject"],
    "Effect":"Allow",
    "Resource":["arn:aws:s3:::companybucket/Development/*"]
 }
```

1. Inicie sesión en la AWS Management Console y abra la consola de Amazon S3 en https://console.aws.amazon.com/s3/.

   Utilice las credenciales de su Cuenta de AWS, no las de usuario de IAM, para iniciar sesión en la consola.

2. Edite la política insertada que creó en el paso anterior.

   a. En el panel de navegación de la izquierda, elija Usuarios.

   b. Elija el nombre de usuario Alice.

   c. En la página de detalles del usuario, elija la pestaña Permisos y expanda la sección Políticas insertadas.

   d. Seleccione Editar política junto al nombre de la política que creó en el paso anterior.

e. Copie la siguiente política y péguela en el campo de texto de la política para sustituir la política existente.

```
{
     "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. Pruebe la política actualizada:

   a. Con el enlace de inicio de sesión del usuario de IAM (consulte Para proporcionar un enlace de inicio de sesión para usuarios de IAM), inicie sesión en la AWS Management Console.

   b. Abra la consola de Amazon S3 en https://console.aws.amazon.com/s3.

   c. En la consola de Amazon S3, compruebe que Alice pueda añadir y descargar un objeto en la carpeta `Development`.

Paso 5.3: Denegar permisos de forma explícita al usuario Alice de IAM para que no obtenga acceso a ninguna otra carpeta del bucket

El usuario Alice ahora puede mostrar el contenido del bucket `companybucket` en el nivel raíz. También puede obtener y colocar objetos en la carpeta `Development`. Si realmente desea ajustar los permisos de acceso, puede denegar de forma explícita el acceso a Alice a cualquier otra carpeta

del bucket. Si existe alguna otra política (política de bucket o ACL) que conceda acceso a Alice a otra carpeta del bucket, esta denegación explícita anula estos permisos.

Puede añadir la siguiente instrucción a la política del usuario Alice que requiere que todas las solicitudes que Alice envía a Amazon S3 incluyan el parámetro `prefix`, cuyo valor puede ser `Development/*` o una cadena vacía.

```
{
   "Sid": "ExplicitlyDenyAnyRequestsForAllOtherFoldersExceptDevelopment",
   "Action": ["s3:ListBucket"],
   "Effect": "Deny",
   "Resource": ["arn:aws:s3:::companybucket"],
   "Condition":{  "StringNotLike": {"s3:prefix":["Development/*",""] },
                  "Null"         : {"s3:prefix":false }
    }
}
```

Existen dos expresiones condicionales en el bloque `Condition`. El resultado de estas expresiones condicionales se combina con el uso de la operación lógica `AND`. Si ambas condiciones son válidas, el resultado de la condición combinada se considera válido. Dado que el `Effect` en esta política es `Deny`, cuando la `Condition` se considera válida, los usuarios no pueden realizar la `Action` especificada.

- La expresión condicional `Null` asegura que las solicitudes de Alice incluyan el parámetro `prefix`.

  El parámetro `prefix` requiere acceso de tipo carpeta. Si envía una solicitud sin el parámetro `prefix`, Amazon S3 devuelve todas las claves de objeto.

  Si la solicitud incluye el parámetro `prefix` con un valor nulo, la expresión se considera válida y, por lo tanto, el parámetro `Condition` también se considera válido. Debe permitir una cadena vacía como valor del parámetro `prefix`. Recuerde que permitir una cadena nula permite a Alice recuperar los elementos del bucket en el nivel raíz como lo hizo la consola en la explicación anterior. Para obtener más información, consulte Paso 4.2: Permitir que los usuarios puedan mostrar el contenido en el nivel raíz de un bucket.
- La expresión condicional `StringNotLike` asegura que la solicitud falle, si se especifica el valor del parámetro `prefix` y no el parámetro `Development/*`.

Siga los pasos de la sección anterior y vuelva a actualizar la política insertada que creó para el usuario Alice.

Copie la siguiente política y péguela en el campo de texto de la política para sustituir la política existente.

```
{
   "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 }
          }
      }
   ]
}
```

Paso 6: Conceder permisos específicos al usuario Bob de IAM

Ahora desea conceder permiso a Bob para trabajar en la carpeta `Finance`. Siga los pasos realizados anteriormente para conceder permisos a Alice, pero sustituya la carpeta `Development`

por la carpeta `Finance`. Para obtener instrucciones paso a paso, consulte Paso 5: Conceder permisos específicos al usuario Alice de IAM.

Paso 7: Proteger la carpeta privada

En este ejemplo, tiene solo dos usuarios. Concedió todos los permisos mínimos requeridos en el nivel de grupo y los permisos en el nivel de usuario solo cuando realmente se requerían permisos en el nivel de usuario individual. Este enfoque ayuda a reducir el esfuerzo para administrar los permisos. A medida que el número de usuarios aumenta, administrar los permisos puede ser complicado. Por ejemplo, no quiere que ninguno de los usuarios de este ejemplo obtenga acceso al contenido de la carpeta `Private`. ¿Cómo asegurarse de no conceder permisos accidentalmente al usuario a la carpeta `Private`? Añada una política que deniegue explícitamente el acceso a la carpeta. Una denegación explícita anula cualquier otro permiso.

Para asegurarse de que la carpeta `Private` sea privada, puede añadir estas dos instrucciones de denegación a la política de grupo:

- Añada la siguiente instrucción para denegar explícitamente cualquier acción sobre los recursos de la carpeta `Private` (`companybucket/Private/*`).

```
{
  "Sid": "ExplictDenyAccessToPrivateFolderToEveryoneInTheGroup",
  "Action": ["s3:*"],
  "Effect": "Deny",
  "Resource":["arn:aws:s3:::companybucket/Private/*"]
}
```

- Además, deniegue el permiso para mostrar la acción de los objetos cuando la solicitud especifica el prefijo `Private/`. Si Bob o Alice abren la carpeta `Private` en la consola, esta política hace que Amazon S3 devuelva una respuesta de error.

```
{
  "Sid": "DenyListBucketOnPrivateFolder",
  "Action": ["s3:ListBucket"],
  "Effect": "Deny",
  "Resource": ["arn:aws:s3:::*"],
  "Condition":{
      "StringLike":{"s3:prefix":["Private/"]}
   }
}
```

Sustituya la política del grupo `Consultants` por una política actualizada que incluya las instrucciones de denegación anteriores. Después de aplicar la política actualizada, ninguno de los usuarios del grupo puede obtener acceso a la carpeta `Private` de su bucket.

1. Inicie sesión en la AWS Management Console y abra la consola de Amazon S3 en https://console.aws.amazon.com/s3/.

   Utilice las credenciales de su Cuenta de AWS, no las de usuario de IAM, para iniciar sesión en la consola.

2. Sustituya la política administrada existente `AllowGroupToSeeBucketListInTheConsole` que se encuentra asociada al grupo `Consultants` por la siguiente política. Recuerde sustituir *`companybucket`* en la política por el nombre de su bucket.

   Para obtener instrucciones, consulte Edición de políticas administradas por el cliente en la Guía del usuario de IAM. Al seguir las instrucciones, asegúrese de seguir las indicaciones para aplicar los cambios a todas las entidades principales a las que la política está asociada.

```
{
  "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/"]}
       }
    }
  ]
}
```

Paso 8: Limpiar

Para realizar una limpieza, abra la consola de IAM y elimine los usuarios Alice y Bob. Para obtener instrucciones paso a paso, consulte Eliminación de un usuario de IAM en la Guía del usuario de IAM.

Para asegurarse de que no le apliquen cargos adicionales por almacenamiento, debe eliminar los objetos y el bucket que creó para este ejercicio.

Recursos relacionados

- Administración de políticas de IAM en la guía del usuario de IAM

Explicaciones que utilizan políticas para administrar el acceso a los recursos de Amazon S3

Este tema proporciona los siguientes ejemplos de tutorial introductorio para ofrecer acceso a los recursos de Amazon S3. Estos ejemplos utilizan la AWS Management Console para crear recursos (buckets, objetos, usuarios) y otorgarles permisos. Luego, los ejemplos le muestran cómo verificar los permisos con las herramientas de línea de comandos para que no tenga que escribir ningún código. Proporcionamos comandos con la AWS Command Line Interface (AWS CLI) y AWS Tools for Windows PowerShell.

- Ejemplo 1: propietario del bucket que concede permisos de bucket a sus usuarios

  Los usuarios de IAM que crea en la cuenta no tienen permisos de forma predeterminada. En este ejercicio, usted le otorga a un usuario permiso para realizar operaciones de bucket y objeto.

- Ejemplo 2: propietario del bucket que concede permisos de bucket entre cuentas

  En este ejercicio, el propietario de un bucket, la cuenta A, concede permisos entre cuentas a otra Cuenta de AWS, la cuenta B. Luego, la cuenta B delega esos permisos a los usuarios de la cuenta.

- Administración de permisos de objetos cuando los propietarios del objeto y del bucket no son los mismos

  Los escenarios de ejemplo en este caso incluyen a un propietario de bucket que concede permisos de objetos a otros, pero no todos los objetos en el bucket le pertenecen al propietario del bucket. ¿Qué permisos necesita el propietario del bucket y cómo puede delegar esos permisos?

  La Cuenta de AWS que crea un bucket se denomina propietario del bucket. El propietario puede conceder permisos a otras Cuentas de AWS para cargar objetos, y las Cuentas de AWS que crean objetos son las propietarias de estos. El propietario del bucket no tiene permisos sobre aquellos objetos creados por otras Cuentas de AWS. Si el propietario del bucket escribe una política de bucket que concede acceso a los objetos, la política no se aplica a los objetos que le pertenecen a otras cuentas.

  En este caso, el propietario del objeto primero debe otorgar permisos al propietario del bucket con una Access Control List (ACL, Lista de control de acceso) de objetos. Luego, el propietario del bucket puede delegar esos permisos de objeto a otros, a usuarios de su propia cuenta o a otra Cuenta de AWS, como se muestra en los siguientes ejemplos.

- Ejemplo 3: propietario del bucket que concede a sus usuarios permisos para objetos que no posee

  En este ejercicio, el propietario del bucket primero obtiene permisos del propietario del objeto. Luego, el propietario del bucket delega esos permisos a usuarios de su propia cuenta.
- Ejemplo 4: Propietario de bucket que concede permisos entre cuentas para objetos que no le pertenecen

  Después de recibir los permisos del propietario del objeto, el propietario del bucket no puede delegar permisos a otras Cuentas de AWS, ya que no se admite la delegación entre cuentas (consulte Delegación de permisos). En lugar de eso, el propietario del bucket puede crear un rol de IAM con permisos para realizar operaciones específicas (como Get object) y permitir que otra Cuenta de AWS asuma ese rol. Luego, cualquiera que asuma el rol puede acceder a los objetos. En este ejemplo se muestra cómo el propietario de un bucket utiliza un rol de IAM para habilitar esta delegación entre cuentas.

Antes de probar los tutoriales de ejemplo

Estos ejemplos utilizan la AWS Management Console para crear recursos y conceder permisos. Para probar los permisos, en los ejemplos se utilizan las herramientas de la línea de comandos, AWS CLI y AWS Tools for Windows PowerShell, por lo que no necesita escribir ningún código. Para probar los permisos, debe configurar una de estas herramientas. Para obtener más información, consulte Configuración de las herramientas para los tutoriales.

Además, cuando se crean recursos, en estos ejemplos no se utilizan credenciales de usuario raíz de una Cuenta de AWS. En lugar de eso, usted crea un usuario administrador en estas cuentas para realizar estas tareas.

Acerca del uso de un usuario administrador para crear recursos y conceder permisos

AWS Identity and Access Management (IAM) recomienda no usar las credenciales de usuario raíz de la Cuenta de AWS para realizar solicitudes. En lugar de eso, cree un usuario o rol de IAM, concédale acceso completo y luego utilice sus credenciales para realizar solicitudes. Recibe el nombre de usuario o rol administrador. Para obtener más información, consulte Credenciales de Usuario raíz de la cuenta de AWS e identidades de IAM en la Referencia general de AWS y Prácticas recomendadas de IAM en la Guía del usuario de IAM.

Todos los tutoriales de ejemplo en esta sección utilizan las credenciales de usuario administrador. Si no creó un usuario administrador para su Cuenta de AWS, en los temas se indica cómo hacerlo.

Para iniciar sesión en la AWS Management Console con las credenciales de usuario, debe utilizar la dirección URL de inicio de sesión de usuario de IAM. La consola de IAM proporciona esta URL para su Cuenta de AWS. Los temas le muestran cómo obtener el URL.

Configuración de las herramientas para los tutoriales

En los ejemplos introductorios (consulte Explicaciones que utilizan políticas para administrar el acceso a los recursos de Amazon S3), utilice la AWS Management Console para crear recursos y conceder permisos. Para probar los permisos, en los ejemplos se utilizan las herramientas de la línea de comandos, AWS Command Line Interface (AWS CLI) y AWS Tools for Windows PowerShell, por lo que no necesita escribir ningún código. Para probar los permisos, debe configurar una de estas herramientas.

Para configurar la AWS CLI

1. Descargue y configure la AWS CLI. Para obtener instrucciones, consulte los siguientes temas en la Guía del usuario de la AWS Command Line Interface:

   Instalación o actualización a la versión más reciente de la AWS Command Line Interface

   Comenzar a utilizar AWS Command Line Interface
2. Configure el perfil predeterminado.

   Puede almacenar las credenciales de usuarios en el archivo de configuración de la AWS CLI. Cree un perfil predeterminado en el archivo de configuración con las credenciales de su Cuenta de AWS. Para obtener instrucciones sobre cómo encontrar y editar el archivo de configuración de AWS CLI, consulte Opciones de los archivos de configuración y credenciales.

   ```
   [default]
   aws_access_key_id = access key ID
   aws_secret_access_key = secret access key
   region = us-west-2
   ```
3. Verifique la configuración introduciendo el siguiente comando en el símbolo del sistema. Ninguno de estos comandos proporciona las credenciales de forma explícita, por lo que se utilizan las credenciales del perfil predeterminado.

   - Pruebe el comando `help`.

     ```
     aws help
     ```

- Para obtener una lista de buckets en la cuenta configurada, utilice el comando `aws s3 ls`.

```
aws s3 ls
```

A medida que avanza por los tutoriales, creará usuarios y guardará las credenciales de usuario en los archivos de configuración mediante la creación de perfiles, como se muestra en el siguiente ejemplo. Estos perfiles tienen los nombres `AccountAadmin` y `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
```

Para ejecutar un comando con estas credenciales de usuario, agregue el parámetro `--profile` especificando el nombre de perfil. El siguiente comando de la AWS CLI recupera una lista de objetos en *`examplebucket`* y especifica el perfil `AccountBadmin`.

```
aws s3 ls s3://examplebucket --profile AccountBadmin
```

Como alternativa, puede configurar un conjunto de credenciales de usuario como el perfil predeterminado cambiando la variable de entorno `AWS_DEFAULT_PROFILE` en el símbolo del sistema. Después de haberlo hecho, siempre que ejecute los comandos de la AWS CLI sin el parámetro `--profile`, la AWS CLI utiliza el perfil que configure en la variable de entorno como perfil predeterminado.

```
$ export AWS_DEFAULT_PROFILE=AccountAadmin
```

Configuración de AWS Tools for Windows PowerShell

1. Descargue y configure la AWS Tools for Windows PowerShell. Para las instrucciones, vaya a Instalación de la AWS Tools for Windows PowerShell en la Guía del usuario de AWS Tools for Windows PowerShell.

> **Note**
>
> Para cargar el módulo de AWS Tools for Windows PowerShell, debe habilitar la ejecución de scripts de PowerShell. Para obtener más información, consulte Habilitación de la ejecución del script en la Guía del usuario de AWS Tools for Windows PowerShell.

2. Para estos tutoriales, debe especificar las credenciales de AWS por sesión con el comando `Set-AWSCredentials`. El comando guarda las credenciales en un almacén persistente (parámetro `-StoreAs`).

```
Set-AWSCredentials -AccessKey AccessKeyID -SecretKey SecretAccessKey -
storeas string
```

3. Verifique la configuración.

   - Para recuperar una lista de comandos disponibles que puede utilizar para las operaciones de Amazon S3, ejecute el comando `Get-Command`.

```
Get-Command -module awspowershell -noun s3* -StoredCredentials string
```

   - Para recuperar una lista de objetos de un bucket, ejecute el comando `Get-S3Object`.

```
Get-S3Object -BucketName bucketname -StoredCredentials string
```

Para obtener una lista de comandos, consulte Referencia de Cmdlet para Herramientas de AWS para PowerShell.

Ahora está listo para probar los tutoriales. Siga los enlaces provistos al principio de cada sección.

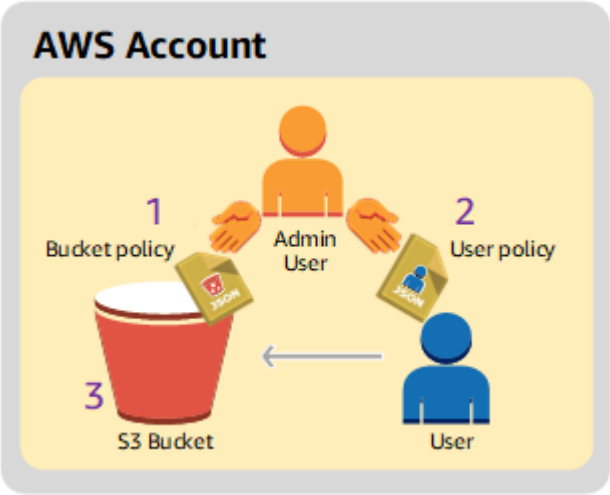
Ejemplo 1: propietario del bucket que concede permisos de bucket a sus usuarios

> **Important**
>
> Conceder permisos a roles de IAM es preferible a conceder permisos a usuarios individuales. Para obtener más información acerca de cómo conceder permisos a roles de IAM, consulte Descripción de permisos entre cuentas y uso de roles de IAM.

Temas

En este tutorial, una Cuenta de AWS es propietaria de un bucket y la cuenta incluye un usuario de IAM. De forma predeterminada, el usuario no tiene permisos. La cuenta principal debe conceder permisos al usuario para realizar cualquier tarea. El propietario del bucket y la cuenta principal son el mismo. Por tanto, para conceder permisos al usuario en el bucket, la Cuenta de AWS puede utilizar una política de bucket, una política de usuario o ambas. El propietario de la cuenta concederá algunos permisos utilizando una política de bucket y otros permisos mediante una política de usuario.

En los siguientes pasos se resume el tutorial:



1. El administrador de la cuenta crea una política de bucket en la que se concede un conjunto de permisos al usuario.

2. El administrador de la cuenta adjunta una política de usuario al usuario, en la que se conceden permisos adicionales.
3. A continuación, el usuario prueba los permisos concedidos tanto mediante la política de bucket como mediante la política de usuario.

Para este ejemplo, necesitará una Cuenta de AWS. En lugar de usar las credenciales de usuario raíz de la cuenta, creará un usuario administrador (consulte Acerca del uso de un usuario administrador para crear recursos y conceder permisos). Nos referimos a la Cuenta de AWS y al usuario administrador como se muestra en la siguiente tabla.

ID de cuenta	Cuenta denominada	Usuario administrador de la cuenta
1111-1111-1111	Cuenta A	AccountAadmin

> **Note**
>
> El usuario administrador de este ejemplo es AccountAdmin, que hace referencia a la cuenta A y no AccountAdmin.

Todas las tareas de creación de usuarios y concesión de permisos se realizan en la AWS Management Console. Para verificar los permisos, en la explicación se utilizan herramientas de la línea de comandos, AWS Command Line Interface (AWS CLI) y AWS Tools for Windows PowerShell, por lo que no necesita escribir código.

Prepararse para el tutorial

1. Asegúrese de tener una Cuenta de AWS que cuente con un usuario con privilegios de administrador.
   a. Si lo necesita, regístrese para obtener una Cuenta de AWS. Nos referiremos a esta cuenta como Cuenta A.
      i. Vaya a https://aws.amazon.com/s3 y elija Crear una cuenta de AWS.
      ii. Siga las instrucciones en pantalla.

AWSCuando la cuenta esté activada y lista para usar, lo notificará por email.

b. En la Cuenta A, cree un usuario administrador **`AccountAadmin`**. Con las credenciales de la Cuenta A, inicie sesión en la consola de IAM y realice los siguientes pasos:

   i. Cree al usuario **`AccountAadmin`** y tenga en cuenta las credenciales de seguridad del mismo.

      Para obtener instrucciones, consulte Creación de un usuario de IAM en la Cuenta de AWS en la Guía del usuario de IAM.

   ii. Conceda privilegios de administrador a AccountAadmin adjuntando una política de usuario que le conceda acceso total.

      Para obtener instrucciones, consulte Administración de políticas de IAM en la Guía del usuario de IAM.

   iii. Tenga en cuenta la URL de inicio de sesión de usuario de IAM para AccountAadmin. Tendrá que usar esta dirección URL al iniciar sesión en la AWS Management Console. Para obtener más información sobre dónde encontrar la URL de inicio de sesión, consulte Iniciar sesión en la AWS Management Console como usuario de IAM en la Guía del usuario de IAM. Tenga en cuenta la URL de cada una de las cuentas.

2. Configure AWS CLI o AWS Tools for Windows PowerShell. Asegúrese de guardar las credenciales del usuario administrador de la siguiente manera:

   - Si utiliza la AWS CLI, cree un perfil, `AccountAadmin`, en el archivo de configuración.
   - Si usa AWS Tools for Windows PowerShell, asegúrese de almacenar las credenciales para la sesión como `AccountAadmin`.

   Para obtener instrucciones, consulte Configuración de las herramientas para los tutoriales.

Paso 1: Creación de recursos en la cuenta A y concesión de permisos

Con las credenciales del usuario `AccountAadmin` en la Cuenta A y la URL especial de inicio de sesión del usuario de IAM, inicie sesión en la AWS Management Console y realice los siguientes pasos:

1. Creación de recursos de un bucket y un usuario de IAM

a. En la consola de Amazon S3, cree un bucket. Tenga en cuenta la Región de AWS en la que creó el bucket. Para obtener instrucciones, consulte Crear un bucket.

b. En la consola de IAM, haga lo siguiente:

   i. Cree un usuario llamado Dave.

      Para obtener instrucciones paso a paso, consulte Creación de usuarios de IAM (consola) en la Guía del usuario de IAM.

   ii. Tenga en cuenta las credenciales de `UserDave`.

   iii. Tenga en cuenta el nombre de recurso de Amazon (ARN) para el usuario Dave. En la consola de IAM, seleccione el usuario y en la pestaña Resumen podrá ver el ARN del usuario.

2. Concesión de permisos.

   Dado que la cuenta propietaria del bucket y la cuenta principal a la que pertenece el usuario coinciden, la Cuenta de AWS puede conceder permisos al usuario mediante una política de bucket, una política de usuario o ambas. En este ejemplo, usará ambas. Si el objeto también es propiedad de la misma cuenta, el propietario del bucket puede conceder permisos de objeto en la política del bucket (o una política de IAM).

   a. En la consola de Amazon S3, asocie la siguiente política de bucket a *`awsexamplebucket1`*.

      La política tiene dos instrucciones:

      - En la primera instrucción se conceden a Dave los permisos de operación en el bucket `s3:GetBucketLocation` y `s3:ListBucket`.
      - En la segunda instrucción se concede el permiso `s3:GetObject`. Dado que la Cuenta A también es propietaria del objeto, el administrador de la cuenta puede conceder el permiso `s3:GetObject`.

      En la instrucción `Principal`, el ARN del usuario es lo que identifica a Dave. Para obtener más información sobre los elementos de las políticas, consulte Políticas y permisos en 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. Cree una política en línea para el usuario Dave utilizando la siguiente política. La política concede a Dave el permiso `s3:PutObject`. Tendrá que actualizar la política proporcionando el nombre del bucket.

```
{
   "Version": "2012-10-17",
   "Statement": [
      {
         "Sid": "PermissionForObjectOperations",
         "Effect": "Allow",
         "Action": [
            "s3:PutObject"
```

```
            ],
            "Resource": [
               "arn:aws:s3:::awsexamplebucket1/*"
            ]
         }
      ]
}
```

Para obtener instrucciones, consulte Administración de políticas de IAM en la Guía del usuario de IAM. Tenga en cuenta que tendrá que iniciar sesión en la consola con las credenciales de la Cuenta A.

Paso 2: Probar los permisos

Compruebe que los permisos funcionan con las credenciales de Dave. Puede utilizar uno de los dos procedimientos siguientes.

Prueba de los permisos mediante la AWS CLI

1. Actualice el archivo de configuración de la AWS CLI agregando el perfil de `UserDaveAccountA` siguiente. Para obtener más información, consulte Configuración de las herramientas para los tutoriales.

   ```
   [profile UserDaveAccountA]
   aws_access_key_id = access-key
   aws_secret_access_key = secret-access-key
   region = us-east-1
   ```

2. Compruebe que Dave pueda realizar las operaciones según los permisos concedidos en la política de usuario. Cargue un objeto de muestra con el comando de la AWS CLI `put-object`.

   El parámetro `--body` en el comando identifica el archivo de origen para cargar. Por ejemplo, si el archivo está en el directorio raíz del disco C: en un equipo Windows, debe especificar `c:\HappyFace.jpg`. El parámetro `--key` brinda el nombre de clave para el objeto.

   ```
   aws s3api put-object --bucket awsexamplebucket1 --key HappyFace.jpg --
   body HappyFace.jpg --profile UserDaveAccountA
   ```

   Ejecute el siguiente comando de la AWS CLI para obtener el objeto.

```
aws s3api get-object --bucket awsexamplebucket1 --key HappyFace.jpg OutputFile.jpg
 --profile UserDaveAccountA
```

Prueba de los permisos mediante la AWS Tools for Windows PowerShell

1. Almacene las credenciales de Dave como `AccountADave`. A continuación, use estas credenciales para PUT y GET un objeto.

   ```
   set-awscredentials -AccessKey AccessKeyID -SecretKey SecretAccessKey -storeas
    AccountADave
   ```

2. Cargue un objeto de muestra con el comando `Write-S3Object` de AWS Tools for Windows PowerShell con las credenciales almacenadas de Dave.

   ```
   Write-S3Object -bucketname awsexamplebucket1 -key HappyFace.jpg -file HappyFace.jpg
    -StoredCredentials AccountADave
   ```

   Descargue el objeto que cargó en el paso anterior.

   ```
   Read-S3Object -bucketname awsexamplebucket1 -key HappyFace.jpg -file Output.jpg -
   StoredCredentials AccountADave
   ```

Ejemplo 2: propietario del bucket que concede permisos de bucket entre cuentas

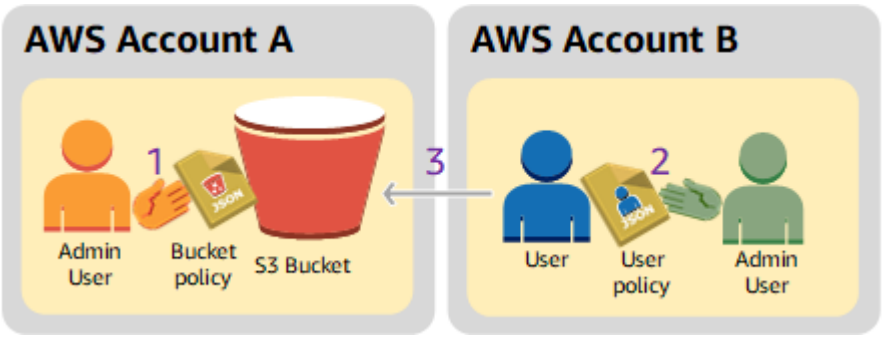
> **Important**
>
> Conceder permisos a roles de IAM es preferible a conceder permisos a usuarios individuales. Para obtener información sobre como hacer esto, consulte Descripción de permisos entre cuentas y uso de roles de IAM.

Temas

Una Cuenta de AWS, por ejemplo, la cuenta A, puede conceder a otra Cuenta de AWS, la cuenta B, permiso para acceder a sus recursos, como buckets y objetos. La cuenta B puede delegar estos permisos a los usuarios en la cuenta. En este escenario de ejemplo, un propietario de bucket concede permiso entre cuentas a otra cuenta para realizar operaciones de bucket específicas.

> **Note**
>
> La cuenta A también puede usar una política de bucket para conceder permisos directamente a un usuario de la cuanta B. Sin embargo, el usuario necesitará permiso de la cuenta principal, la cuenta B, a la que pertenece el usuario, incluso aunque la cuenta B no tenga permisos de la cuenta A. El usuario podrá acceder al recurso siempre y cuando tenga permiso del propietario del recurso y la cuenta principal.

A continuación, se muestra un resumen de los pasos del tutorial:



1. El usuario administrador de la cuenta A adjunta una política de bucket que concede permisos entre cuentas a la cuenta B para realizar operaciones de bucket específicas.

   Tenga en cuenta que el usuario administrador en la cuenta B heredará automáticamente los permisos.
2. El usuario administrador de la cuenta B le adjunta la política de usuario al usuario que delega los permisos que recibió de la cuenta A.
3. El usuario de la cuenta B luego accede a un objeto en el bucket que pertenece a la cuenta A para verificar los permisos.

Para este ejemplo, necesita dos cuentas. En la siguiente tabla se muestra cómo nos referimos a estas cuentas y a sus usuarios administradores. De acuerdo con las directrices de IAM (consulte Acerca del uso de un usuario administrador para crear recursos y conceder permisos), no utilizamos las credenciales de usuario raíz en este tutorial. En lugar de eso, usted crea un usuario administrador en cada cuenta y utiliza esas credenciales cuando se crean recursos y se conceden permisos.

ID de Cuenta de AWS	Cuenta denominada	Usuario administrador de la cuenta
1111-1111-1111	Cuenta A	AccountAadmin
2222-2222-2222	Cuenta B	AccountBadmin

Todas las tareas de creación de usuarios y concesión de permisos se realizan en la AWS Management Console. Para verificar los permisos, en la explicación se utilizan herramientas de línea de comandos, AWS Command Line Interface (CLI) y AWS Tools for Windows PowerShell, por lo que no necesita escribir código.

Prepararse para el tutorial

1. Asegúrese de tener dos Cuentas de AWS y que cada cuenta tenga un usuario administrador, como se muestra en la tabla de la sección anterior.

   a. Si lo necesita, regístrese para obtener una Cuenta de AWS.

   b. Con las credenciales de la cuenta A, inicie sesión en la consola de IAM para crear el usuario administrador:

      i. Cree al usuario **`AccountAadmin`** y tenga en cuenta las credenciales de seguridad. Para obtener instrucciones, consulte Creación de un usuario de IAM en la Cuenta de AWS en la Guía del usuario de IAM.

      ii. Conceda privilegios de administrador a AccountAadmin adjuntando una política de usuario que le conceda acceso total. Para obtener instrucciones, consulte Uso de políticas en la guía del usuario de IAM.

   c. Mientras se encuentra en la consola de IAM, tenga en cuenta la URL de inicio de sesión de usuario de IAM en el Panel. Todos los usuarios de la cuenta deben utilizar esta dirección URL para iniciar sesión en la AWS Management Console.

      Para obtener más información, consulte Cómo los usuarios inician sesión en la cuenta en la Guía del usuario de IAM.

   d. Repita el paso anterior con las credenciales de la cuenta B y cree el usuario administrador **`AccountBadmin`**.

2. Configure AWS Command Line Interface (AWS CLI) o AWS Tools for Windows PowerShell. Asegúrese de guardar las credenciales del usuario administrador de la siguiente manera:

   - Si está usando la AWS CLI, cree dos perfiles, `AccountAadmin` y `AccountBadmin`, en el archivo de configuración.
   - Si usa AWS Tools for Windows PowerShell, asegúrese de almacenar las credenciales para la sesión como `AccountAadmin` y `AccountBadmin`.

   Para obtener instrucciones, consulte Configuración de las herramientas para los tutoriales.

3. Guarde las credenciales de usuario administrador, también denominadas perfiles. Puede utilizar el nombre de perfil en lugar de especificar las credenciales para cada comando que escribe. Para obtener más información, consulte Configuración de las herramientas para los tutoriales.

a. Agregue perfiles en el archivo de credenciales de la AWS CLI para cada uno de los usuarios administradores, `AccountAadmin` y `AccountBadmin` en las dos cuentas.

```
[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 usa AWS Tools for Windows PowerShell, ejecute el siguiente comando.

```
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
```

Paso 1: Realizar las tareas de la cuenta A

Paso 1.1: Iniciar sesión en la AWS Management Console

Con la URL de inicio de sesión de usuario de IAM para la cuenta A, primero inicie sesión en la AWS Management Console como el usuario AccountAadmin. Este usuario creará un bucket y le asociará una política.

Paso 1.2: Crear un bucket

1. En la consola de Amazon S3, cree un bucket. En este ejercicio se asume que el bucket se crea en la Región de AWS Este de EE. UU. (Norte de Virginia) y se llama *amzn-s3-demo-bucket*.

   Para obtener instrucciones, consulte Crear un bucket.

2. Cargue un objeto de ejemplo en el bucket.

   Para obtener instrucciones, consulte Paso 2: Cargar un objeto en el bucket.

Paso 1.3: Asociar una política de bucket para conceder permisos entre cuentas a la cuenta B

La política de bucket concede los permisos de `s3:GetLifecycleConfiguration` y `s3:ListBucket` a la cuenta B. Se asume que aún sigue registrado en la consola con las credenciales de usuario AccountAadmin.

1. Adjunte la siguiente política de bucket a *amzn-s3-demo-bucket*. La política le concede a la cuenta B permiso para las acciones `s3:GetLifecycleConfiguration` y `s3:ListBucket`.

   Para obtener instrucciones, consulte Agregar una política de bucket mediante la consola de 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:::amzn-s3-demo-bucket"
            ]
         }
      ]
   }
   ```

2. Asegúrese de que la cuenta B (y por lo tanto el usuario administrador) pueda realizar las operaciones.

   - Verificación mediante la AWS CLI

     ```
     aws s3 ls s3://amzn-s3-demo-bucket --profile AccountBadmin
     aws s3api get-bucket-lifecycle-configuration --bucket amzn-s3-demo-bucket --
     profile AccountBadmin
     ```

   - Verificación mediante la AWS Tools for Windows PowerShell

```
get-s3object -BucketName amzn-s3-demo-bucket -StoredCredentials AccountBadmin
get-s3bucketlifecycleconfiguration -BucketName amzn-s3-demo-bucket -
StoredCredentials AccountBadmin
```

Paso 2: Realizar las tareas de la cuenta B

Ahora, el administrador de la cuenta B crea un usuario, Dave, y delega los permisos que recibió de la cuenta A.

Paso 2.1: Iniciar sesión en la AWS Management Console

Con el URL de inicio de sesión de usuario de IAM para la cuenta B, primero inicie sesión en la AWS Management Console con el usuario AccountBadmin.

Paso 2.2: Crear el usuario Dave en la cuenta B

En la Consola de IAM, cree un usuario, **`Dave`**.

Para obtener instrucciones, consulte Creación de usuarios de IAM (consola) en la Guía del usuario de IAM.

Paso 2.3: Delegar permisos al usuario Dave

Cree una política en línea para el usuario Dave utilizando la siguiente política. Tendrá que proporcionar el nombre del bucket para actualizar la política.

Se asume que se ha registrado en la consola con las credenciales de usuario AccountBadmin.

```
{
   "Version": "2012-10-17",
   "Statement": [
      {
         "Sid": "Example",
         "Effect": "Allow",
         "Action": [
            "s3:ListBucket"
         ],
         "Resource": [
            "arn:aws:s3:::amzn-s3-demo-bucket"
         ]
      }
   ]
```

```
}
```

Para obtener instrucciones, consulte Administración de políticas de IAM en la Guía del usuario de IAM.

Paso 2.4: Probar los permisos

Ahora, Dave en la cuenta B puede indicar los contenidos de *amzn-s3-demo-bucket* que pertenece a la cuenta A. Puede usar cualquiera de los siguientes procedimientos para verificar los permisos.

Prueba de los permisos mediante la AWS CLI

1. Agregue el perfil `UserDave` al archivo de configuración de la AWS CLI. Para obtener más información acerca del archivo de configuración, consulte Configuración de las herramientas para los tutoriales.

   ```
   [profile UserDave]
   aws_access_key_id = access-key
   aws_secret_access_key = secret-access-key
   region = us-east-1
   ```

2. En el símbolo del sistema, ingrese el siguiente comando de la AWS CLI para asegurarse de que Dave ya pueda obtener una lista de objetos de *amzn-s3-demo-bucket* que pertenece a la cuenta A. Tenga en cuenta que el comando especifica el perfil `UserDave`.

   ```
   aws s3 ls s3://amzn-s3-demo-bucket --profile UserDave
   ```

   Dave no tiene ningún otro permiso. Por lo tanto, si intenta cualquier otra operación, como la siguiente configuración `get-bucket-lifecycle`, Amazon S3 deniega el permiso.

   ```
   aws s3api get-bucket-lifecycle-configuration --bucket amzn-s3-demo-bucket --profile
    UserDave
   ```

Prueba de los permisos mediante AWS Tools for Windows PowerShell

1. Almacene las credenciales de Dave como `AccountBDave`.

   ```
   set-awscredentials -AccessKey AccessKeyID -SecretKey SecretAccessKey -storeas
    AccountBDave
   ```

2. Pruebe el comando List Bucket.

```
get-s3object -BucketName amzn-s3-demo-bucket -StoredCredentials AccountBDave
```

Dave no tiene ningún otro permiso. Por lo tanto, si intenta cualquier otra operación, como la siguiente `get-s3bucketlifecycleconfiguration`, Amazon S3 deniega el permiso.

```
get-s3bucketlifecycleconfiguration -BucketName amzn-s3-demo-bucket -
StoredCredentials AccountBDave
```

Paso 3: (Opcional) Intentar denegar explícitamente

Puede tener permisos concedidos mediante el uso de una lista de control de acceso (ACL), una política de bucket o una política de usuario. Pero si se establece una denegación explícita a través de una política de bucket o una política de usuario, la denegación explícita tiene preferencia sobre cualquier otro permiso. Para las pruebas, actualice la política de bucket y niegue explícitamente los permisos de `s3:ListBucket` a la cuenta B. La política también concede el permiso de `s3:ListBucket`. Sin embargo, la denegación explícita prevalece, y la cuenta B o los usuarios de la cuenta B no podrán mostrar los objetos en *amzn-s3-demo-bucket*.

1. Con las credenciales de usuario `AccountAadmin` en la cuenta A, sustituya la política de bucket por lo siguiente.

```
{
   "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:::amzn-s3-demo-bucket"
         ]
      },
```

```
      {
         "Sid": "Deny permission",
         "Effect": "Deny",
         "Principal": {
            "AWS": "arn:aws:iam::AccountB-ID:root"
         },
         "Action": [
            "s3:ListBucket"
         ],
         "Resource": [
            "arn:aws:s3:::amzn-s3-demo-bucket"
         ]
      }
   ]
}
```

2. Ahora, si intenta obtener una lista de bucket con las credenciales de `AccountBadmin`, se le denegará el acceso.

   - Con la AWS CLI, ejecute el siguiente comando:

     ```
     aws s3 ls s3://amzn-s3-demo-bucket --profile AccountBadmin
     ```

   - Con la AWS Tools for Windows PowerShell, ejecute el siguiente comando:

     ```
     get-s3object -BucketName amzn-s3-demo-bucket -StoredCredentials AccountBDave
     ```

Paso 4: Limpiar

1. Después de que haya terminado de probar, puede realizar lo siguiente para limpiar:

   - Inicie sesión en la AWS Management Console (AWS Management Console) con las credenciales de la cuenta A y haga lo siguiente:

     - En la consola de Amazon S3, elimine la política de bucket asociada a *amzn-s3-demo-bucket*. En Propiedades del bucket, elimine la política en la sección Permisos.
     - Si se creó el bucket para este ejercicio, en la consola de Amazon S3, elimine los objetos y luego elimine el bucket.
     - En la consola de IAM, quite el usuario `AccountAadmin`.

2. Inicie sesión en la consola de IAM con las credenciales de la cuenta B. Elimine el usuario `AccountBadmin`. Para obtener instrucciones paso a paso, consulte Eliminación de un usuario de IAM en la Guía del usuario de IAM.

Ejemplo 3: propietario del bucket que concede a sus usuarios permisos para objetos que no posee

> **Important**
>
> Conceder permisos a roles de IAM es preferible a conceder permisos a usuarios individuales. Para obtener información sobre como hacer esto, consulte Descripción de permisos entre cuentas y uso de roles de IAM.

Temas

- Paso 0: Prepararse para el tutorial
- Paso 1: Realizar las tareas de la cuenta A
- Paso 2: Realizar las tareas de la cuenta B
- Paso 3: Probar los permisos
- Paso 4: Limpiar

El escenario para este ejemplo es que el propietario de un bucket desea conceder permiso para acceder a objetos, pero no todos los objetos del bucket pertenecen al propietario del bucket. Para este ejemplo, el propietario del bucket intenta conceder permiso a usuarios en su propia cuenta.

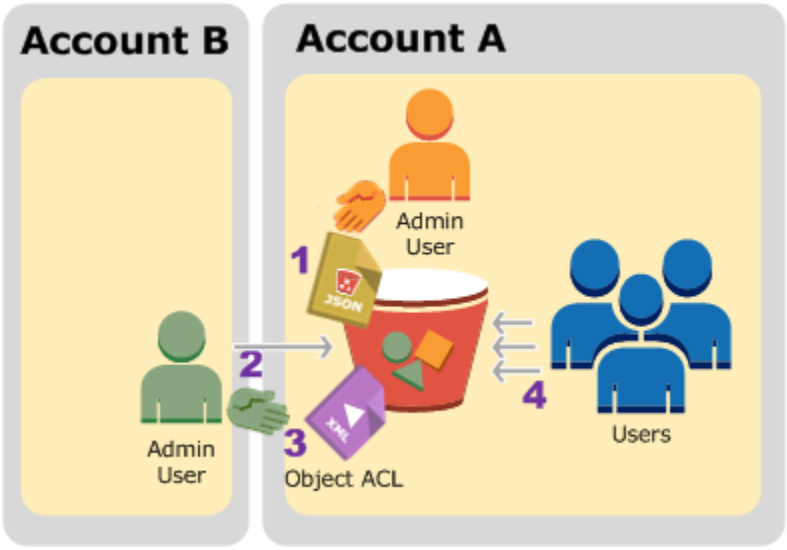
El propietario del bucket puede habilitar otras Cuentas de AWS para cargar objetos. De forma predeterminada, el propietario del bucket no posee objetos escritos en un bucket por otra Cuenta de AWS. Los objetos pertenecen a las cuentas que los escriben en un bucket de S3. Si el propietario del bucket no posee objetos en el bucket, el propietario del objeto primero debe conceder permiso al propietario del bucket con una lista de control de acceso (ACL) del objeto. A continuación, el propietario del bucket puede conceder permisos a un objeto que no posee. Para obtener más información, consulte Propiedad de los buckets y objetos de Amazon S3.

Si el propietario del bucket aplica la configuración de propietario del bucket obligatorio de S3 Object Ownership para el bucket, el propietario del bucket será el propietario de todos los objetos del bucket, incluidos los objetos escritos por otra Cuenta de AWS. Este enfoque resuelve el problema de

que los objetos no pertenecen al propietario del bucket. A continuación, puede delegar permisos a los usuarios de su propia cuenta o a otras Cuentas de AWS.

> **Note**
>
> S3 Object Ownership es una configuración de bucket de Amazon S3 que puede usar para controlar la propiedad de los objetos que se cargan en el bucket y para activar o desactivar las ACL. De forma predeterminada, la propiedad de objetos se establece en la configuración impuesta por el propietario del bucket. Además, todas las ACL están deshabilitadas. Cuando las ACL están deshabilitadas, el propietario del bucket posee todos los objetos del bucket y administra su acceso de forma exclusiva mediante políticas de administración de acceso. La mayoría de los casos de uso modernos de Amazon S3 ya no requieren el uso de ACL. Le recomendamos desactivar las ACL, excepto en circunstancias inusuales en las que necesite controlar el acceso a cada objeto de manera individual. Si las ACL están desactivadas, puede usar políticas para controlar el acceso a todos los objetos del bucket, independientemente de quién haya subido los objetos al bucket. Para obtener más información, consulte Control de la propiedad de los objetos y desactivación de las ACL del bucket.

En este ejemplo, asumimos que el propietario del bucket no ha aplicado la configuración de propietario del bucket obligatorio de Object Ownership. El propietario del bucket delega permiso a usuarios en su propia cuenta. A continuación, se muestra un resumen de los pasos del tutorial:



1. El usuario administrador de la cuenta A adjunta una política de bucket con dos instrucciones.

- Conceda permiso entre cuentas a la cuenta B para cargar objetos.
- Permita que un usuario en su propia cuenta pueda acceder a los objetos en el bucket.

2. El usuario administrador de la cuenta B carga objetos al bucket que pertenece a la cuenta A.
3. El administrador de la cuenta B actualiza la ACL de objetos y le concede al propietario del bucket permiso de control total sobre el objeto.
4. El usuario de la cuenta A lo verifica al acceder a los objetos en el bucket, independientemente de quién es el propietario.

Para este ejemplo, necesita dos cuentas. La siguiente tabla muestra cómo denominamos a estas cuentas y a los usuarios administradores en estas cuentas. En este tutorial, no utilizará las credenciales de usuario raíz de la cuenta, de acuerdo con las directrices recomendadas de IAM. Para obtener más información, consulte Acerca del uso de un usuario administrador para crear recursos y conceder permisos. En cambio, usted crea un administrador en cada cuenta y usa esas credenciales para crear recursos y conceder permisos.

ID de Cuenta de AWS	Cuenta denominada	Administrador en la cuenta
1111-1111-1111	Cuenta A	AccountAadmin
2222-2222-2222	Cuenta B	AccountBadmin

Todas las tareas de creación de usuarios y concesión de permisos se realizan en la AWS Management Console. Para verificar los permisos, en la explicación se utilizan herramientas de la línea de comandos, AWS Command Line Interface (AWS CLI) y AWS Tools for Windows PowerShell, por lo que no necesita escribir código.

Paso 0: Prepararse para el tutorial

1. Asegúrese de tener dos Cuentas de AWS y que cada cuenta tenga un administrador, como se muestra en la tabla de la sección anterior.

   a. Si lo necesita, regístrese para obtener una Cuenta de AWS.

   b. Con las credenciales de la cuenta A, inicie sesión en la consola de IAM y haga lo siguiente para crear un usuario administrador:

- Cree al usuario **`AccountAadmin`** y tenga en cuenta las credenciales de seguridad del mismo. Para obtener más información sobre cómo agregar usuarios, consulte Creación de un usuario de IAM en su Cuenta de AWS en la Guía del usuario de IAM.
- Conceda permisos de administrador a AccountAadmin al asociar una política de usuario que conceda acceso completo. Para obtener instrucciones, consulte Administración de políticas de IAM en la Guía del usuario de IAM.
- En el panel de la consola de IAM, tenga en cuenta la URL de inicio de sesión de usuario de IAM. Los usuarios de esta cuenta deben usar esta dirección URL para iniciar sesión en la AWS Management Console. Para obtener más información, consulte Cómo los usuarios inician sesión en la cuenta en la Guía del usuario de IAM.

c. Repita el paso anterior con las credenciales de la cuenta B y cree el usuario administrador **`AccountBadmin`**.

2. Configure AWS CLI o Tools for Windows PowerShell. Asegúrese de guardar las credenciales de administrador de la siguiente manera:

   - Si está usando la AWS CLI, cree dos perfiles, `AccountAadmin` y `AccountBadmin`, en el archivo de configuración.
   - Si utiliza Tools for Windows PowerShell, asegúrese de almacenar las credenciales de la sesión como `AccountAadmin` y `AccountBadmin`.

   Para obtener instrucciones, consulte Configuración de las herramientas para los tutoriales.

Paso 1: Realizar las tareas de la cuenta A

Realice los siguientes pasos para la cuenta A:

Paso 1.1: Iniciar sesión en la consola

Con la URL de inicio de sesión de usuario de IAM para la cuenta A, inicie sesión en la AWS Management Console como el usuario **`AccountAadmin`**. Este usuario creará un bucket y le asociará una política.

Paso 1.2: Crear un bucket y un usuario y agregar una política de bucket para conceder permisos de usuario

1. En la consola de Amazon S3, cree un bucket. En este ejercicio se supone que el bucket se crea en la Región de AWS Este de EE. UU. (Norte de Virginia) y el nombre es *amzn-s3-demo-bucket1*.

   Para obtener instrucciones, consulte Crear un bucket.

2. En la Consola de IAM, cree un usuario, **Dave**.

   Para obtener instrucciones paso a paso, consulte Creación de usuarios de IAM (consola) en la Guía del usuario de IAM.

3. Tenga en cuenta las credenciales del usuario Dave.

4. En la consola de Amazon S3, adjunte la siguiente política de bucket al bucket *amzn-s3-demo-bucket1*. Para obtener instrucciones, consulte Agregar una política de bucket mediante la consola de Amazon S3. Siga los pasos para añadir una política de bucket. Para obtener información acerca de cómo buscar ID de cuenta, consulte Búsqueda de su ID de Cuenta de AWS.

   La política le concede a la cuenta B permisos para `s3:PutObject` y `s3:ListBucket`. La política también le concede al usuario `Dave` el permiso `s3:GetObject`.

```
{
   "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:::amzn-s3-demo-bucket1/*",
            "arn:aws:s3:::amzn-s3-demo-bucket1"
         ]
      },
```

```
      {
         "Sid": "Statement3",
         "Effect": "Allow",
         "Principal": {
            "AWS": "arn:aws:iam::AccountA-ID:user/Dave"
         },
         "Action": [
            "s3:GetObject"
         ],
         "Resource": [
            "arn:aws:s3:::amzn-s3-demo-bucket1/*"
         ]
      }
   ]
}
```

Paso 2: Realizar las tareas de la cuenta B

Ahora que la cuenta B tiene permisos para realizar operaciones en el bucket de la cuenta A, el administrador de la cuenta B hace lo siguiente:

- Carga un objeto en el bucket de la cuenta A
- Agrega un permiso en la ACL de objetos para concederle el control total al propietario del bucket de la cuenta A

Uso de la AWS CLI

1. Mediante el uso del comando `put-object` de la AWS CLI, cargue un objeto. El parámetro `--body` en el comando identifica el archivo de origen para cargar. Por ejemplo, si el archivo está en la unidad `C:` de un equipo Windows, debe especificar `c:\HappyFace.jpg`. El parámetro `--key` brinda el nombre de clave para el objeto.

   ```
   aws s3api put-object --bucket amzn-s3-demo-bucket1 --key HappyFace.jpg --body
    HappyFace.jpg --profile AccountBadmin
   ```

2. Añada un permiso a la ACL de objetos para concederle al propietario del bucket el control total del objeto. Para obtener información acerca de cómo encontrar un ID de usuario canónico, consulte Buscar el ID de usuario canónico de su Cuenta de AWS en la Guía de referencia de la Administración de cuentas de AWS.

```
aws s3api put-object-acl --bucket amzn-s3-demo-bucket1 --key HappyFace.jpg --grant-
full-control id="AccountA-CanonicalUserID" --profile AccountBadmin
```

Uso de Tools for Windows PowerShell

1. Mediante el uso del comando `Write-S3Object`, cargue un objeto.

```
Write-S3Object -BucketName amzn-s3-demo-bucket1 -key HappyFace.jpg -file
 HappyFace.jpg -StoredCredentials AccountBadmin
```

2. Añada un permiso a la ACL de objetos para concederle al propietario del bucket el control total del objeto.

```
Set-S3ACL -BucketName amzn-s3-demo-bucket1 -Key HappyFace.jpg -CannedACLName
 "bucket-owner-full-control" -StoredCreden
```

Paso 3: Probar los permisos

Ahora verifique que el usuario Dave en la cuenta A pueda acceder al objeto propiedad de la cuenta B.

Uso de la AWS CLI

1. Agregue las credenciales del usuario Dave al archivo de configuración de la AWS CLI y cree un nuevo perfil, `UserDaveAccountA`. Para obtener más información, consulte Configuración de las herramientas para los tutoriales.

```
[profile UserDaveAccountA]
aws_access_key_id = access-key
aws_secret_access_key = secret-access-key
region = us-east-1
```

2. Ejecute el comando `get-object` CLI para descargarlo `HappyFace.jpg` y guardarlo localmente. Para proporcionar las credenciales del usuario Dave, debe añadir el parámetro `--profile`.

```
aws s3api get-object --bucket amzn-s3-demo-bucket1 --key
 HappyFace.jpg Outputfile.jpg --profile UserDaveAccountA
```

Uso de Tools for Windows PowerShell

1. Almacene las credenciales del usuario Dave de AWS, como `UserDaveAccountA`, en el almacén persistente.

```
Set-AWSCredentials -AccessKey UserDave-AccessKey -SecretKey UserDave-
SecretAccessKey -storeas UserDaveAccountA
```

2. Ejecute el `Read-S3Object` comando para descargar el `HappyFace.jpg` objeto y guárdelo localmente. Para proporcionar las credenciales del usuario Dave, debe añadir el parámetro `-StoredCredentials`.

```
Read-S3Object -BucketName amzn-s3-demo-bucket1 -Key HappyFace.jpg -file
 HappyFace.jpg  -StoredCredentials UserDaveAccountA
```

Paso 4: Limpiar

1. Después de que haya terminado de probar, puede realizar lo siguiente para limpiar:

   - Inicie sesión en la AWS Management Console con las credenciales de la cuenta A y haga lo siguiente:

     - En la consola de Amazon S3, elimine la política de bucket asociada a *amzn-s3-demo-bucket1*. En Propiedades del bucket, elimine la política en la sección Permisos.
     - Si se creó el bucket para este ejercicio, en la consola de Amazon S3, elimine los objetos y luego elimine el bucket.
     - En la consola de IAM, quite el usuario AccountAadmin. Para obtener instrucciones paso a paso, consulte Eliminación de un usuario de IAM en la Guía del usuario de IAM.

2. Inicie sesión en la AWS Management Console con las credenciales de la cuenta B. En la consola de IAM, elimine el usuario AccountBadmin.

Ejemplo 4: Propietario de bucket que concede permisos entre cuentas para objetos que no le pertenecen

Temas

En este ejemplo de escenario, usted posee un bucket y ha permitido que otras Cuentas de AWS carguen objetos. Si ha aplicado la configuración de propietario del bucket obligatorio de S3 Object Ownership para el bucket, será propietario de todos los objetos del bucket, incluidos los objetos escritos por otra Cuenta de AWS. Este enfoque resuelve el problema de que los objetos no le pertenecen, el propietario del bucket. A continuación, puede delegar permisos a los usuarios de su propia cuenta o a otras Cuentas de AWS. Supongamos que la configuración de propietario del bucket obligatorio de S3 Object Ownership no está habilitada. Es decir, el bucket puede tener objetos que pertenecen a otras Cuentas de AWS.

Suponga que como propietario del bucket debe conceder permiso entre cuentas para ciertos objetos, independientemente de quién sea el propietario, a un usuario en otra cuenta. Por ejemplo, ese usuario podría ser una aplicación de facturación que necesita obtener acceso a los metadatos de los objetos. Hay dos cuestiones clave:

- El propietario del bucket no tiene permisos sobre aquellos objetos creados por otras Cuentas de AWS. Para que el propietario del bucket conceda permisos para objetos que no le pertenecen, el propietario del objeto primero debe conceder permiso al propietario del bucket. El propietario del objeto es la Cuenta de AWS que creó los objetos. Luego, el propietario del bucket puede delegar esos permisos.
- La cuenta propietaria del bucket delega permisos a usuarios en su propia cuenta (consulte Ejemplo 3: propietario del bucket que concede a sus usuarios permisos para objetos que no posee). Sin embargo, la cuenta del propietario del bucket no puede delegar permisos en otra Cuentas de AWS porque no se admite la delegación entre cuentas.

En este caso, el propietario del bucket puede crear un rol de AWS Identity and Access Management (IAM) con permiso para acceder a los objetos. A continuación, el propietario del bucket puede conceder permiso a otra Cuenta de AWS para asumir el rol, habilitarlo de manera temporal para acceder a los objetos en el bucket.

> **Note**
>
> S3 Object Ownership es una configuración de bucket de Amazon S3 que puede usar para controlar la propiedad de los objetos que se cargan en el bucket y para activar o desactivar las ACL. De forma predeterminada, la propiedad de objetos se establece en la configuración impuesta por el propietario del bucket. Además, todas las ACL están deshabilitadas. Cuando las ACL están deshabilitadas, el propietario del bucket posee todos los objetos del bucket y administra su acceso de forma exclusiva mediante políticas de administración de acceso. La mayoría de los casos de uso modernos de Amazon S3 ya no requieren el uso de ACL. Le recomendamos desactivar las ACL, excepto en circunstancias inusuales en las que necesite controlar el acceso a cada objeto de manera individual. Si las ACL están desactivadas, puede usar políticas para controlar el acceso a todos los objetos del bucket, independientemente de quién haya subido los objetos al bucket. Para obtener más información, consulte Control de la propiedad de los objetos y desactivación de las ACL del bucket.

Descripción de permisos entre cuentas y uso de roles de IAM

Las funciones de IAM hacen posible diferentes situaciones en las que se delega acceso a los recursos y el acceso a otras cuentas es una de las situaciones clave. En este ejemplo, el propietario del bucket, la cuenta A, usa un rol de IAM para delegar de forma temporal el acceso a los objetos entre cuentas a usuarios de otra Cuenta de AWS, la cuenta C. Cada rol de IAM que crea trae adjuntas las siguientes dos políticas:

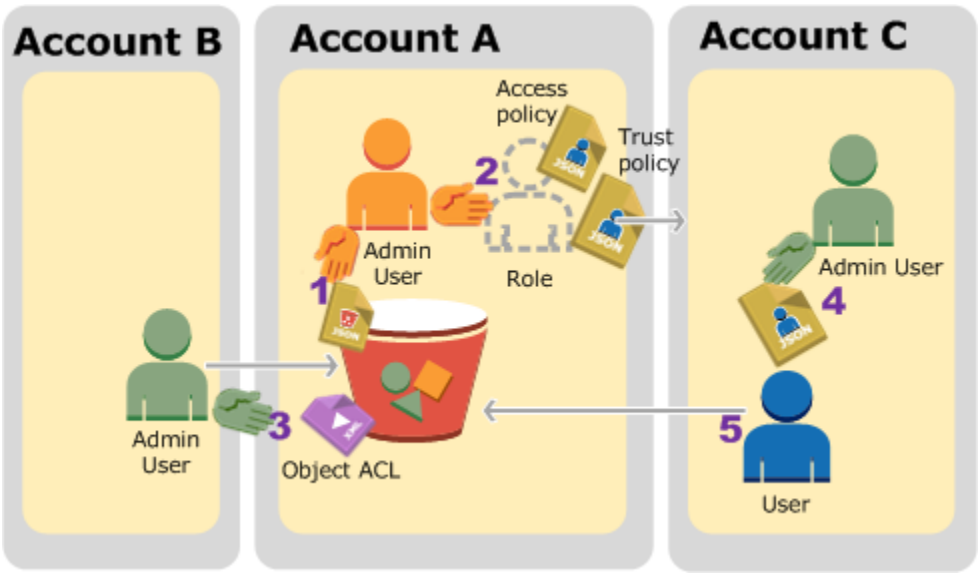
- Una política de confianza que identifica a otra Cuenta de AWS para que pueda asumir el rol.
- Una política de acceso que define qué permisos, por ejemplo, `s3:GetObject`, se conceden cuando alguien asume la función. Para ver una lista de permisos que puede especificar en una política, consulte Acciones de políticas para Amazon S3.

La Cuenta de AWS que se identifica en la política de confianza luego otorga permiso a su usuario para asumir el rol. El usuario puede luego realizar lo siguiente para obtener acceso a los objetos:

- Asumir la función y, como respuesta, obtener las credenciales de seguridad temporales
- Con las credenciales de seguridad temporales, obtener acceso a los objetos en el bucket

Para obtener más información acerca de los roles de IAM, consulte Roles de IAM en la guía del usuario de IAM.

A continuación, se muestra un resumen de los pasos del tutorial:



1. El usuario administrador de la cuenta A asocia una política de bucket por la cual concede un permiso condicional a la cuenta B para cargar objetos.
2. El administrador de la cuenta A crea un rol de IAM, con la cual establece una relación de confianza con la cuenta C, de manera que los usuarios de esa cuenta puedan obtener acceso a la cuenta A. La política de acceso asociada con la función limita lo que el usuario de la cuenta C puede hacer cuando el usuario obtiene acceso a la cuenta A.
3. El administrador de la cuenta B carga un objeto al bucket que pertenece a la cuenta A y concede un permiso de control absoluto al propietario del bucket.
4. El administrador de la cuenta C crea un usuario y asocia una política de usuario que le permite al usuario asumir la función.
5. El usuario de la cuenta C primero asume la función, la cual le devuelve al usuario credenciales de seguridad temporales. Con esas credenciales temporales, el usuario obtiene acceso a los objetos en el bucket.

Para este ejemplo, se necesitan tres cuentas. La siguiente tabla muestra cómo denominamos a estas cuentas y a los usuarios administradores en estas cuentas. De acuerdo con las directrices de IAM (consulte Acerca del uso de un usuario administrador para crear recursos y conceder permisos), no utilizamos las credenciales de Usuario raíz de la cuenta de AWS en este tutorial. En lugar de eso, usted crea un usuario administrador en cada cuenta y utiliza esas credenciales cuando se crean recursos y se conceden permisos.

ID de Cuenta de AWS	Cuenta denominada	Usuario administrador de la cuenta
1111-1111-1111	Cuenta A	AccountAadmin
2222-2222-2222	Cuenta B	AccountBadmin
3333-3333-3333	Cuenta C	AccountCadmin

Paso 0: Prepararse para el tutorial

> **Note**
>
> Puede que quiera abrir un editor de textos y tomar notas a medida que se explican los pasos. Específicamente, necesitará ID de cuentas, ID de usuarios canónicos, URL de inicio de sesión de usuarios de IAM para cada cuenta a fin de conectarse a la consola y Amazon Resource Name (ARN, Nombres de recursos de Amazon) de los usuarios de IAM y roles.

1. Asegúrese de tener tres Cuentas de AWS y que cada cuenta tenga un usuario administrador, como se muestra en la tabla de la sección anterior.

   a. Regístrese para obtener Cuentas de AWS, según sea necesario. Denominamos estas cuentas cuenta A, cuenta B y cuenta C.

   b. Con las credenciales de la cuenta A, inicie sesión en la consola de IAM y haga lo siguiente para crear un usuario administrador:

      - Cree al usuario **`AccountAadmin`** y tenga en cuenta sus credenciales de seguridad. Para obtener más información sobre cómo agregar usuarios, consulte Creación de un usuario de IAM en su Cuenta de AWS en la Guía del usuario de IAM.

- Conceda privilegios de administrador a AccountAadmin adjuntando una política de usuario que le conceda acceso total. Para obtener instrucciones, consulte Administración de políticas de IAM en la Guía del usuario de IAM.
- En el panel de la consola de IAM, tenga en cuenta la URL de inicio de sesión de usuario de IAM. Los usuarios de esta cuenta deben usar esta dirección URL para iniciar sesión en la AWS Management Console. Para obtener más información, consulte Inicio de sesión en la AWS Management Console como usuario de IAM en la Guía del usuario de IAM.

c. Repita el paso anterior para crear usuarios administradores de la cuenta B y de la cuenta C.

2. Para la cuenta C, tenga en cuenta el ID de usuario canónico.

   Cuando crea un rol de IAM en la cuenta A, la política de confianza le concede a la cuenta C el permiso para asumir la función mediante la especificación del ID de la cuenta. Puede buscar la información de la cuenta de la siguiente manera:

   a. Utilice el ID de la Cuenta de AWS o el alias de la cuenta, el nombre de usuario de IAM y la contraseña para iniciar sesión en la consola de Amazon S3.

   b. Elija el nombre de un bucket de Amazon S3 para ver los detalles de dicho bucket.

   c. Elija la pestaña Permissions (Permisos) y, a continuación, elija Access Control List (Lista de control de acceso).

   d. En la sección Acceso para la Cuenta de AWS, en la columna Cuenta, hay un identificador largo, por ejemplo, `c1daexampleaaf850ea79cf0430f33d72579fd1611c97f7ded193374c0b163b6`. Este es el ID de usuario canónico.

3. Cuando cree una política de bucket, necesitará la siguiente información. Tenga en cuenta estos valores:

   - El ID de usuario canónico de la cuenta A: cuando el administrador de la cuenta A concede el permiso condicional para cargar objetos al administrador de la cuenta B, la condición especifica el ID de usuario canónico del usuario de la cuenta A que debe obtener control absoluto de los objetos.

     > **Note**
     >
     > El ID de usuario canónico es un concepto exclusivo de Amazon S3. Es la versión ofuscada de 64 caracteres del ID de la cuenta.

- ARN de usuario para el administrador de la cuenta B: puede encontrar el ARN del usuario en la consola de IAM. Debe seleccionar el usuario y buscar el ARN del usuario en la pestaña Resumen.

  En la política de bucket, se concede permiso al `AccountBadmin` para cargar objetos y se usa el ARN para especificar el usuario. El siguiente es un ejemplo de valor de ARN:

  ```
  arn:aws:iam::AccountB-ID:user/AccountBadmin
  ```

4. Configure AWS Command Line Interface (CLI) o AWS Tools for Windows PowerShell. Asegúrese de guardar las credenciales del usuario administrador de la siguiente manera:

   - Si está usando la AWS CLI, cree perfiles, `AccountAadmin` y `AccountBadmin`, en el archivo de configuración.
   - Si usa AWS Tools for Windows PowerShell, asegúrese de almacenar las credenciales para la sesión como `AccountAadmin` y `AccountBadmin`.

   Para obtener instrucciones, consulte Configuración de las herramientas para los tutoriales.

Paso 1: Realizar las tareas de la cuenta A

En este ejemplo, la cuenta A es la propietaria del bucket. Por lo tanto, el usuario AccountAadmin en la cuenta A hará lo siguiente:

- Crear un bucket.
- Asocie una política de bucket que conceda el permiso al administrador de la cuenta B para cargar objetos.
- Cree un rol de IAM que conceda a la cuenta C el permiso para asumir el rol, de manera que pueda acceder a los objetos en el bucket.

Paso 1.1: Iniciar sesión en la AWS Management Console

Con la URL de inicio de sesión de usuario de IAM para la cuenta A, primero inicie sesión en la AWS Management Console como el usuario **`AccountAadmin`**. Este usuario creará un bucket y le asociará una política.

Paso 1.2: Crear un bucket y asociar una política de bucket

En la consola de Amazon S3, haga lo siguiente:

1. Cree un bucket. Para este ejercicio, se supone que el nombre del bucket es *amzn-s3-demo-bucket1*.

   Para obtener instrucciones, consulte Crear un bucket.

2. Adjunte la siguiente política de bucket. La política concede permiso condicional al permiso de administrador de la cuenta B para cargar objetos.

   Actualice la política proporcionando sus propios valores para *amzn-s3-demo-bucket1*, *AccountB-ID* y *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:::amzn-s3-demo-bucket1/*"
        },
        {
            "Sid": "112",
            "Effect": "Deny",
            "Principal": {
                "AWS": "arn:aws:iam::AccountB-ID:user/AccountBadmin"
            },
            "Action": "s3:PutObject",
            "Resource": "arn:aws:s3:::amzn-s3-demo-bucket1/*",
            "Condition": {
                "StringNotEquals": {
                    "s3:x-amz-grant-full-control": "id=CanonicalUserId-of-
AWSaccountA-BucketOwner"
                }
            }
        }
    ]
```

```
}
```

Paso 1.3: Creación de un rol de IAM para permitir a la cuenta C el acceso entre cuentas en la cuenta A

En la Consola de IAM, cree un rol de IAM (**`examplerole`**) que conceda permiso a la cuenta C para asumir el rol. Asegúrese de que sigue registrado como administrador de la cuenta A, ya que el rol se debe crear en la cuenta A.

1. Antes de crear el rol, prepare la política administrada que define los permisos que requiere el rol. Más tarde, en otro paso, la asociará al rol.

   a. En el panel de navegación de la izquierda, elija Políticas y, a continuación, elija Crear política.

   b. Junto a Create Your Own Policy, seleccione Select.

   c. Escriba **`access-accountA-bucket`** en el campo Nombre de la política.

   d. Copie la siguiente política de acceso y péguela en el campo Documento de la política. La política de acceso le concede al rol el permiso de `s3:GetObject`, de manera que cuando el usuario de la cuenta C asuma el rol, solo pueda realizar la operación `s3:GetObject`.

   ```
   {
     "Version": "2012-10-17",
     "Statement": [
       {
         "Effect": "Allow",
         "Action": "s3:GetObject",
         "Resource": "arn:aws:s3:::amzn-s3-demo-bucket1/*"
       }
     ]
   }
   ```

   e. Seleccione Crear política.

   La nueva política aparece en la lista de políticas administradas.

2. En el panel de navegación de la izquierda, elija Roles y, a continuación, elija Creación de rol nuevo.

3. En Seleccionar tipo de rol, seleccione Rol para acceso entre cuentas y, luego, elija el botón Seleccionar ubicado al lado de Proporcionar acceso entre Cuentas de AWS de su propiedad.

4. Escriba el ID de la cuenta C.

   Para este tutorial, no es obligatorio que los usuarios tengan la autenticación multifactor (MFA) para asumir el rol, por lo que deje esa opción desmarcada.
5. Seleccione Next Step para establecer los permisos que se asociarán al rol.
6. Seleccione la casilla de verificación situada junto a la política access-accountA-bucket que creó y, a continuación, elija Siguiente paso.

   Aparece la página Review para que confirme las configuraciones de la función antes de crearla. Un elemento muy importante para tener en cuenta en esta página es el enlace que puede enviar a los usuarios que necesitan usar esta función. Los usuarios que hacen clic en el enlace se dirigen directamente a la página Cambio de rol con los campos ID de cuenta y Nombre de rol ya completados. También puede ver este enlace más tarde, en la página Role Summary de cualquier rol con permisos entre cuentas.
7. Escriba `examplerole` para el nombre del rol y, luego, elija Siguiente paso.
8. Después de revisar el rol, elija Crear rol.

   La función `examplerole` se muestra en la lista de funciones.
9. Elija el nombre del rol `examplerole`.
10. Seleccione la pestaña Relaciones de confianza.
11. Elija Mostrar el documento de política y verifique que la política de confianza que se muestra coincide con la siguiente política.

    La siguiente política de confianza establece una relación de confianza con la cuenta C mediante la concesión del permiso para realizar la acción `sts:AssumeRole`. Para obtener más información, consulte AssumeRole en la Referencia de la API de AWS Security Token Service.

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Sid": "",
      "Effect": "Allow",
      "Principal": {
        "AWS": "arn:aws:iam::AccountC-ID:root"
      },
      "Action": "sts:AssumeRole"
    }
```

```
  ]
}
```

12. Tenga en cuenta el nombre de recurso de Amazon (ARN) del rol `examplerole` que creó.

    Más adelante en los pasos siguientes, se asocia una política de usuario para permitirle a un usuario de IAM que asuma esta función y la función se identifica con el valor de ARN.

Paso 2: Realizar las tareas de la cuenta B

El bucket de ejemplo que pertenece a la cuenta A necesita objetos que pertenezcan a otras cuentas. En este paso, el administrador de la cuenta B carga un objeto con las herramientas de línea de comandos.

- Mediante el uso del comando `put-object` de la AWS CLI, cargue un objeto en *`amzn-s3-demo-bucket1`*.

  ```
  aws s3api put-object --bucket amzn-s3-demo-bucket1 --key HappyFace.jpg --
  body HappyFace.jpg --grant-full-control id="canonicalUserId-ofTheBucketOwner" --
  profile AccountBadmin
  ```

  Tenga en cuenta lo siguiente:

  - El parámetro `--Profile` especifica el perfil `AccountBadmin`, por lo que el objeto pertenece a la cuenta B.
  - El parámetro `grant-full-control` le concede al propietario del bucket permiso de control absoluto sobre el objeto según lo requiere la política de bucket.
  - El parámetro `--body` identifica el archivo de origen para cargar. Por ejemplo, si el archivo está en la unidad C: de un equipo con Windows, se especifica `c:\HappyFace.jpg`.

Paso 3: Realización de las tareas de la cuenta C

En los pasos anteriores, la cuenta A ya ha creado un rol, `examplerole`, mediante el cual se establece una relación de confianza con la cuenta C. Este rol permite a los usuarios de la cuenta C acceder a la cuenta A. En este paso, el administrador de la cuenta C crea un usuario (Dave) y le delega el permiso `sts:AssumeRole` que recibió de la cuenta A. Este enfoque le permite a Dave asumir `examplerole` y obtener acceso temporal a la cuenta A. La política de acceso que la

cuenta A asoció al rol limita lo que Dave puede hacer cuando accede a la cuenta A, específicamente, obtener objetos en *amzn-s3-demo-bucket1*.

Paso 3.1: Creación de un usuario en la cuenta C y delegación de un permiso para asumir examplerole

1. Con la URL de inicio de sesión de usuario de IAM para la cuenta C, primero inicie sesión en la AWS Management Console como el usuario **AccountCadmin**.

2. En la consola de IAM, cree el usuario Dave.

   Para obtener instrucciones paso a paso, consulte Creación de usuarios de IAM (AWS Management Console) en la Guía del usuario de IAM.

3. Tenga en cuenta las credenciales de Dave. Dave necesitará estas credenciales para asumir la función `examplerole`.

4. Cree una política integrada para el usuario de IAM Dave a fin de delegar el permiso `sts:AssumeRole` a Dave en el rol `examplerole` en la cuenta A.

   a. En el panel de navegación de la izquierda, elija Usuarios.

   b. Elija el nombre de usuario Dave.

   c. En la página de detalles del usuario, seleccione la pestaña Permisos y, luego, expanda la sección Políticas insertadas.

   d. Seleccione hacer clic aquí (o Crear política de usuario).

   e. Elija Custom Policy y después Select.

   f. Escriba un nombre para la política en el campo Nombre de la política.

   g. Copie la siguiente política en el campo Documento de la política.

      Debe actualizar la política proporcionando el *AccountA-ID*.

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": ["sts:AssumeRole"],
      "Resource": "arn:aws:iam::AccountA-ID:role/examplerole"
    }
  ]
```

```
}
```

h. Seleccione Apply Policy.

5. Guarde las credenciales de Dave en el archivo de configuración de la AWS CLI mediante la agregación de otro perfil, `AccountCDave`.

```
[profile AccountCDave]
aws_access_key_id = UserDaveAccessKeyID
aws_secret_access_key = UserDaveSecretAccessKey
region = us-west-2
```

Paso 3.2: Asunción del rol (examplerole) y acceso a los objetos

Ahora Dave puede obtener acceso a los objetos en el bucket que pertenecen a la cuenta A de la siguiente manera:

- Dave primero asume la función `examplerole` con sus propias credenciales. Esto devuelve credenciales temporales.
- Con las credenciales temporales, Dave obtiene acceso a los objetos en el bucket de la cuenta A.

1. En el símbolo del sistema, ejecute el comando `assume-role` de la AWS CLI con el perfil `AccountCDave`.

   Debe actualizar el valor de ARN en el comando proporcionando el *AccountA-ID* donde se define `examplerole`.

```
aws sts assume-role --role-arn arn:aws:iam::AccountA-ID:role/examplerole --profile
 AccountCDave --role-session-name test
```

   Como respuesta, AWS Security Token Service (AWS STS) muestra credenciales de seguridad temporales (ID de clave de acceso, clave de acceso secreta y token de sesión).

2. Guarde las credenciales de seguridad temporales en el archivo de configuración de la AWS CLI en el perfil `TempCred`.

```
[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. En el símbolo del sistema, ejecute el siguiente comando de la AWS CLI para acceder a los objetos con las credenciales temporales. Por ejemplo, el comando especifica la Application Programming Interface (API, Interfaz de programación de aplicaciones) de head-object para recuperar los metadatos de los objetos para el objeto `HappyFace.jpg`.

```
aws s3api get-object --bucket amzn-s3-demo-bucket1 --
key HappyFace.jpg SaveFileAs.jpg --profile TempCred
```

Como la política de acceso asociada con la función `examplerole` permite las acciones, Amazon S3 procesa la solicitud. Puede probar cualquier otra acción en cualquier otro objeto del bucket.

Si prueba cualquier otra acción, por ejemplo, `get-object-acl`, se le negará el permiso porque no se le permite esa acción al rol.

```
aws s3api get-object-acl --bucket amzn-s3-demo-bucket1 --key HappyFace.jpg --
profile TempCred
```

Usamos el usuario Dave para asumir la función y obtener acceso al objeto con credenciales temporales. También podría ser una aplicación en la cuenta C que obtuviera acceso a los objetos en el bucket *amzn-s3-demo-bucket1*. La aplicación puede obtener las credenciales de seguridad temporales y la cuenta C puede delegar el permiso de la aplicación para asumir la función `examplerole`.

Paso 4: Limpiar

1. Después de que haya terminado de probar, puede realizar lo siguiente para limpiar:

   - Inicie sesión en la AWS Management Console con las credenciales de la cuenta A y haga lo siguiente:

     - En la consola de Amazon S3, elimine la política de bucket asociada a *amzn-s3-demo-bucket1*. En Propiedades del bucket, elimine la política en la sección Permisos.
     - Si se creó el bucket para este ejercicio, en la consola de Amazon S3, elimine los objetos y luego elimine el bucket.

- En la consola de IAM, elimine el `examplerole` que ha creado en la cuenta A. Para obtener instrucciones paso a paso, consulte Eliminación de un usuario de IAM en la Guía del usuario de IAM.
- En la consola de IAM, quite el usuario AccountAadmin.

2. Inicie sesión en la consola de IAM con las credenciales de la cuenta B. Elimine el usuario AccountBadmin.
3. Inicie sesión en la consola de IAM con las credenciales de la cuenta C. Elimine AccountCadmin y el usuario Dave.

Recursos relacionados

Para obtener más información relacionada con este tutorial, consulte los siguientes recursos en la Guía del usuario de IAM:

- Creación de un rol para delegar permisos a un usuario de IAM
- Tutorial: Delegación del acceso entre Cuentas de AWS mediante roles de IAM
- Administración de políticas de IAM

Cómo autoriza Amazon S3 una solicitud

Cuando Amazon S3 recibe una solicitud, por ejemplo, una operación de bucket o de objeto, primero verifica que el solicitante tenga los permisos necesarios. Para decidir si autoriza la solicitud o no, Amazon S3 evalúa todas las políticas de acceso, las políticas de usuario y las políticas basadas en recursos (política de bucket, lista de control de acceso (ACL) de bucket y ACL de objeto) relevantes.

> **Note**
>
> Si la comprobación de permisos de Amazon S3 no encuentra permisos válidos, se devuelve un error de permiso denegado de acceso denegado (403 Prohibido). Para obtener más información, consulte Solución de errores de acceso denegado (403 Prohibido) en Amazon S3.

Para determinar si el solicitante tiene permiso para realizar la operación específica, Amazon S3 hace lo siguiente, por orden, cuando recibe una solicitud:

1. Convierte todas las políticas de acceso relevantes (política de usuario, política de bucket y ACL) en tiempo de ejecución en un conjunto de políticas para evaluación.
2. Evalúa el conjunto de políticas resultante en los siguiente pasos. En cada paso, Amazon S3 evalúa un subconjunto de políticas en un contexto específico, en función de la autoridad del contexto.
   a. Contexto de usuario: en el contexto de usuario, la cuenta principal a la que pertenece el usuario es la autoridad del contexto.

      Amazon S3 evalúa un subconjunto de políticas perteneciente a la cuenta principal. Este subconjunto incluye la política de usuario que la cuenta principal asocia al usuario. Si la cuenta principal también es propietaria del recurso en la solicitud (bucket u objeto), Amazon S3 también evalúa las políticas de recursos correspondientes (política de bucket, ACL de bucket y ACL de objeto) al mismo tiempo.

      El usuario debe tener permiso de la cuenta principal para realizar la operación.

      Este paso se aplica solo si la solicitud la realiza un usuario en una Cuenta de AWS. Si la solicitud se realiza con las credenciales de usuario raíz de una Cuenta de AWS, Amazon S3 omite este paso.

b. Contexto de bucket: en el contexto de bucket, Amazon S3 evalúa las políticas que pertenecen a la Cuenta de AWS que posee el bucket.

   Si la solicitud es para una operación de bucket, el solicitante debe tener permiso del propietario del bucket. Si la solicitud es para un objeto, Amazon S3 evalúa todas las políticas pertenecientes al propietario del bucket para verificar si el propietario del bucket no denegó explícitamente el acceso al objeto. Si se estableció una denegación explícita, Amazon S3 no autoriza la solicitud.

c. Contexto de objeto: si la solicitud es para un objeto, Amazon S3 evalúa el subconjunto de políticas perteneciente al propietario del objeto.

A continuación, se presentan algunos escenarios de ejemplo que ilustran cómo Amazon S3 autoriza una solicitud.

Example - El solicitante es una entidad principal de IAM

Si el solicitante es una entidad principal de IAM, Amazon S3 debe determinar si la Cuenta de AWS principal a la que pertenece esa entidad principal le concedió a esta el permiso necesario para realizar la operación. Además, si la solicitud es para una operación de bucket, como una solicitud para mostrar el contenido del bucket, Amazon S3 debe verificar que el propietario del bucket le haya otorgado permiso al solicitante para realizar la operación. Para realizar una operación específica en un recurso, la entidad principal de IAM necesita permiso tanto de la Cuenta de AWS principal a la que pertenece como de la Cuenta de AWS a la que pertenece el recurso.

Example - El solicitante es una entidad principal de IAM: en caso de que sea una solicitud para una operación en un objeto que el propietario del bucket no posee

Si la solicitud es para una operación en un objeto que no pertenece al propietario del bucket, además de asegurarse de que el solicitante tenga permisos del propietario del objeto, Amazon S3 también debe revisar la política de bucket para asegurarse de que el propietario del bucket no haya establecido la denegación explícita en el objeto. El propietario del bucket (que paga la factura) puede denegar explícitamente el acceso a los objetos del bucket independientemente de quién sea el propietario. El propietario del bucket también puede eliminar cualquier objeto del bucket.

De forma predeterminada, cuando otra Cuenta de AWS carga un objeto en el bucket de S3, esa cuenta (el escritor del objeto) es propietario del objeto, tiene acceso a él y puede conceder acceso a él a otros usuarios a través de listas de control de acceso (ACL). Puede utilizar la propiedad de

objetos para cambiar este comportamiento predeterminado de modo que las ACL estén desactivadas y, como propietario del bucket, tenga automáticamente la propiedad de todos los objetos del bucket. Como resultado, el control de acceso de los datos se basa en políticas, tales como las políticas de usuario de IAM, las políticas de bucket de S3, las políticas de puntos de conexión de nube privada virtual (VPC) y las políticas de control de servicios (SCP) de AWS Organizations. Para obtener más información, consulte Control de la propiedad de los objetos y desactivación de las ACL del bucket.

Para obtener más información acerca de cómo Amazon S3 evalúa las políticas de acceso para autorizar o denegar solicitudes de operaciones de buckets y operaciones con objetos, consulte los siguientes temas:

Temas

- Cómo hace Amazon S3 para autorizar una solicitud para una operación de bucket
- Cómo hace Amazon S3 para autorizar una solicitud para una operación con objetos

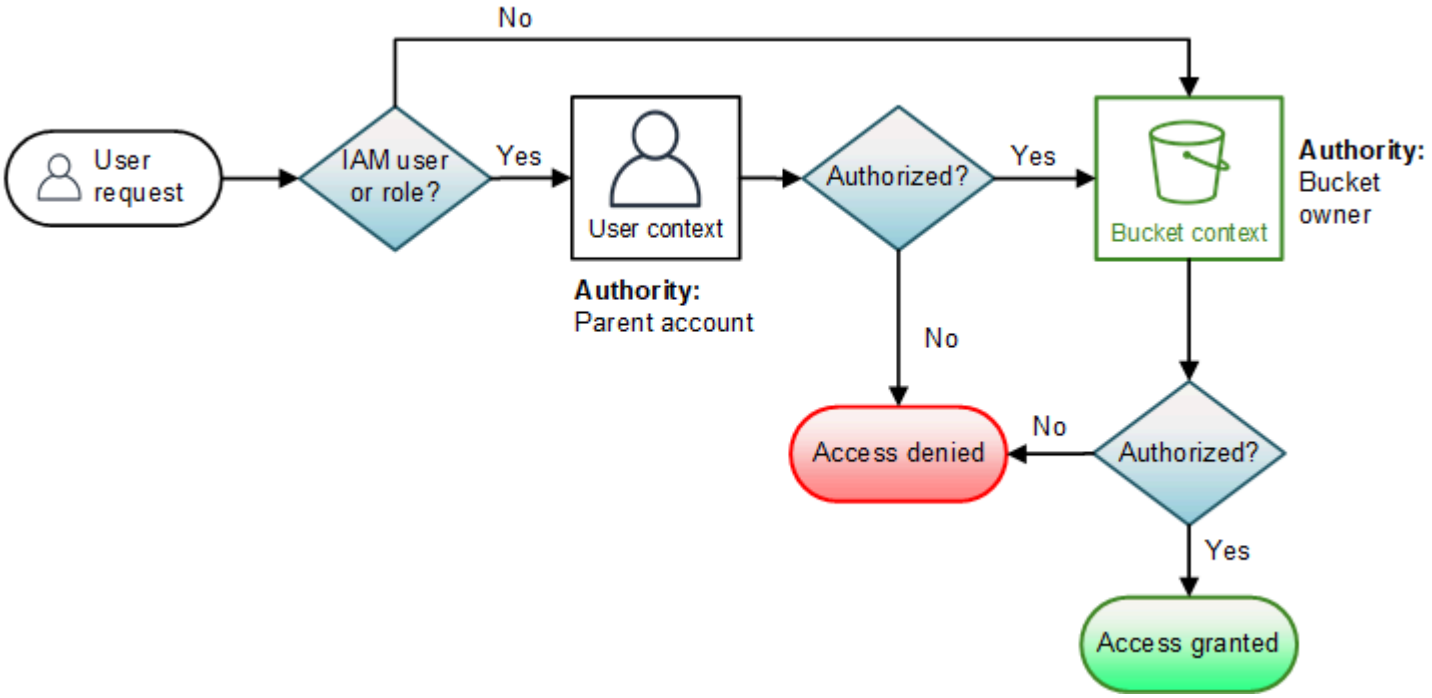
Cómo hace Amazon S3 para autorizar una solicitud para una operación de bucket

Cuando Amazon S3 recibe una solicitud para una operación de bucket, Amazon S3 convierte todos los permisos pertinentes en un conjunto de políticas para evaluar en el tiempo de ejecución. Los permisos pertinentes incluyen permisos basados en recursos (por ejemplo: políticas de bucket y listas de control de acceso a bucket) y políticas de usuario si la solicitud procede de una entidad principal de IAM. A continuación, Amazon S3 evalúa el conjunto de políticas resultante en una serie de pasos de acuerdo con un contexto específico: contexto de usuario o contexto de bucket:

1. Contexto de usuario: si el solicitante es una entidad principal de IAM, esta debe tener permiso de la Cuenta de AWS principal a la que pertenece. En este paso, Amazon S3 evalúa un subconjunto de políticas perteneciente a la cuenta principal (también denominada autoridad del contexto). Este subconjunto de políticas incluye la política de usuario que la cuenta principal asocia al principal. Si la cuenta principal también es propietaria del recurso en la solicitud (en este caso, el bucket), Amazon S3 también evalúa las políticas de recursos correspondientes (política de bucket y ACL de bucket) al mismo tiempo. Siempre que se realiza una solicitud de operación de bucket, los registros de acceso al servidor registran el ID canónico del solicitante. Para obtener más información, consulte Registro de solicitudes con registro de acceso al servidor.
2. Contexto de bucket: el solicitante debe tener permisos del propietario del bucket para realizar una operación de bucket específica. En este paso, Amazon S3 evalúa un subconjunto de políticas que pertenecen a la Cuenta de AWS que es propietaria del bucket.

El propietario del bucket puede conceder permisos mediante una política de bucket o una ACL de bucket. Si la Cuenta de AWS que es propietaria del bucket también es la cuenta principal de una entidad principal de IAM, esta puede configurar los permisos del bucket en una política de usuario.

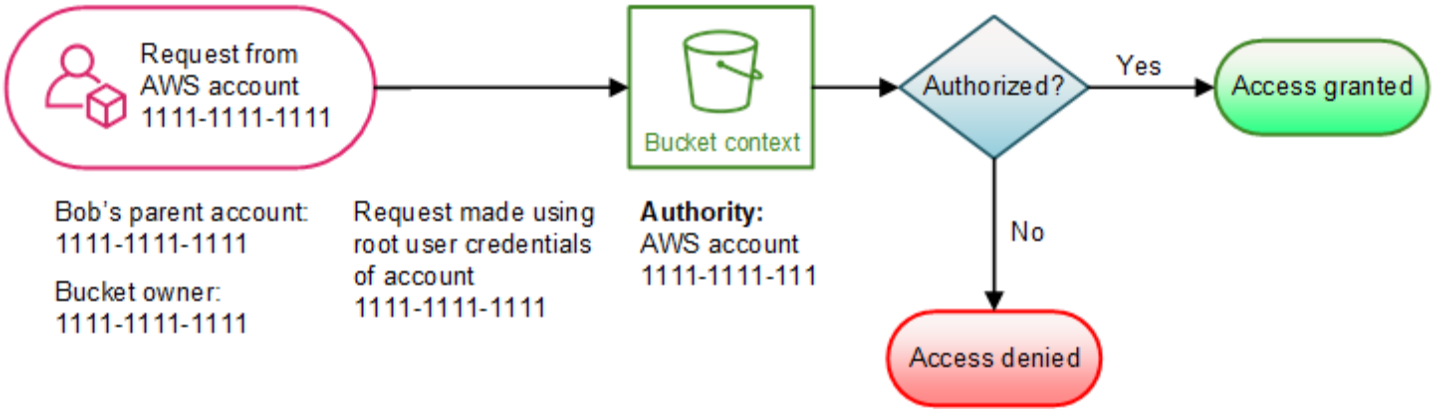
A continuación se muestra una ilustración gráfica de la evaluación de una operación de bucket basada en el contexto.



Los siguientes ejemplos ilustran la lógica de evaluación.

Ejemplo 1: operación de bucket solicitada por el propietario del bucket

En este ejemplo, el propietario del bucket envía una solicitud para una operación de bucket mediante el uso de credenciales raíz de la Cuenta de AWS.

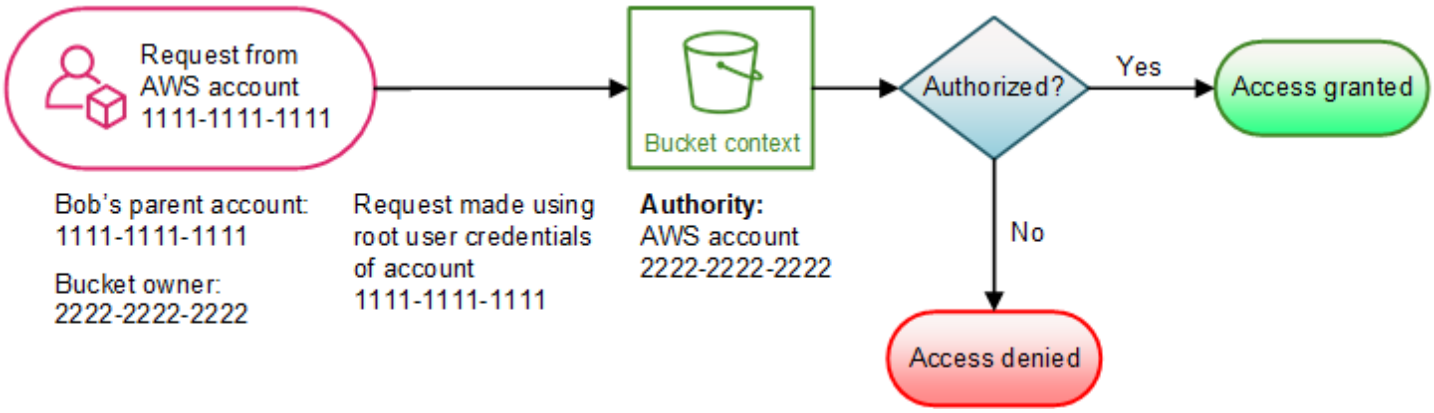


Amazon S3 realiza la evaluación de contexto de la siguiente manera:

1. Dado que la solicitud se realiza con las credenciales de usuario raíz de una Cuenta de AWS, el contexto de usuario no se evalúa.
2. En el contexto de bucket, Amazon S3 revisa la política de bucket para determinar si el solicitante tiene permiso para realizar la operación. Amazon S3 autoriza la solicitud.

Ejemplo 2: operación de bucket solicitada por una Cuenta de AWS que no es propietaria del bucket

En este ejemplo, se realiza una solicitud con credenciales de usuario raíz de la Cuenta de AWS 1111-1111-1111 para una operación de bucket perteneciente a la Cuenta de AWS 2222-2222-2222. No hay ningún usuario de IAM involucrado en esta solicitud.

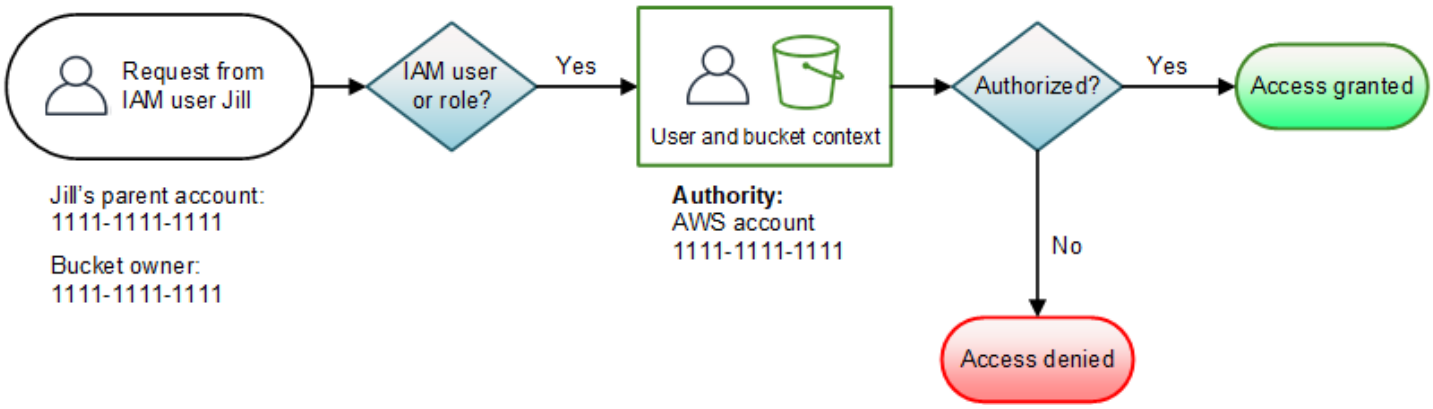


En este ejemplo, Amazon S3 evalúa el contexto de la siguiente manera:

1. Dado que la solicitud se realiza con las credenciales de usuario raíz de una Cuenta de AWS, el contexto de usuario no se evalúa.
2. En el contexto de bucket, Amazon S3 examina la política de bucket. Si el propietario del bucket (Cuenta de AWS 2222-2222-2222) no autorizó a la Cuenta de AWS 1111-1111-1111 a realizar la operación solicitada, Amazon S3 deniega la solicitud. De lo contrario, Amazon S3 acepta la solicitud y realiza la operación.

Ejemplo 3: operación de bucket solicitada por una entidad principal de IAM cuya Cuenta de AWS principal también es propietaria del bucket

En el ejemplo, la solicitud la envía Jill, una usuaria de IAM en la Cuenta de AWS 1111-1111-1111, que también es propietaria del bucket.



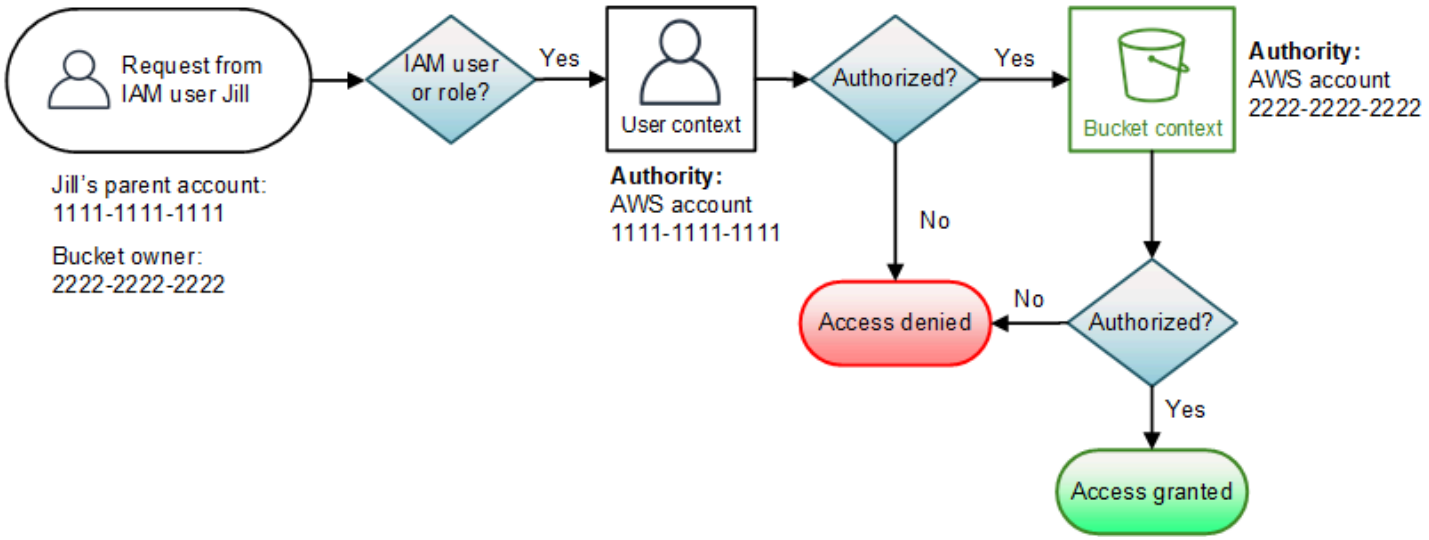
Amazon S3 realiza la siguiente evaluación de contexto:

1. Dado que la solicitud surge de una entidad principal de IAM, en el contexto de usuario, Amazon S3 evalúa todas las políticas que pertenecen a la Cuenta de AWS principal para determinar si Jill cuenta con el permiso necesario para realizar la operación.

   En este ejemplo, la Cuenta de AWS principal 1111-1111-1111 a la que pertenece la entidad principal también es propietaria del bucket. Como consecuencia, además de la política de usuario, Amazon S3 también evalúa la política de bucket y la ACL de bucket en el mismo contexto porque pertenecen a la misma cuenta.
2. Dado que Amazon S3 evaluó la política de bucket y la ACL de bucket como parte del contexto de usuario, no evalúa el contexto de bucket.

Ejemplo 4: operación de bucket solicitada por una entidad principal de IAM cuya Cuenta de AWS principal no es propietaria del bucket

En este ejemplo, la solicitud la envía Jill, una usuaria de IAM cuya Cuenta de AWS principal es 1111-1111-1111, pero el bucket es propiedad de otra Cuenta de AWS, 2222-2222-2222.



Jill necesitará contar con los permisos correspondientes de la Cuenta de AWS principal y el propietario del bucket. Amazon S3 evalúa el contexto de la siguiente manera:

1. Dado que la solicitud es de un principal de IAM, Amazon S3 evalúa el contexto de usuario mediante la revisión de las políticas estipuladas por la cuenta para verificar que Jill tenga los permisos necesarios. Si Jill tiene permiso, Amazon S3 continúa con la evaluación del contexto de bucket. Si Jill no tiene el permiso, deniega la solicitud.
2. En el contexto de bucket, Amazon S3 verifica que el propietario del bucket 2222-2222-2222 haya otorgado permiso a Jill (o a su Cuenta de AWS principal) para realizar la operación solicitada. Si Jill tiene ese permiso, Amazon S3 acepta la solicitud y realiza la operación. De lo contrario, Amazon S3 deniega la solicitud.

Cómo hace Amazon S3 para autorizar una solicitud para una operación con objetos

Cuando Amazon S3 recibe una solicitud para una operación de bucket, convierte todos los permisos relevantes —permisos basados en recursos (lista de control de acceso [ACL] de objeto, política de bucket, ACL de bucket) y las políticas de usuario de IAM— en un conjunto de políticas para evaluar en tiempo de ejecución. Luego, evalúa el conjunto de políticas resultante en una serie de pasos. En cada paso, evalúa un subconjunto de políticas en tres contextos específicos: contexto de usuario, contexto de bucket y contexto de objeto:

1. Contexto de usuario: si el solicitante es una entidad principal de IAM, esta debe tener permiso de la Cuenta de AWS principal a la que pertenece. En este paso, Amazon S3 evalúa un subconjunto de políticas perteneciente a la cuenta principal (también denominada autoridad del contexto). Este subconjunto de políticas incluye la política de usuario que la cuenta principal asocia al principal.

Si la cuenta principal también es propietaria del recurso en la solicitud (bucket u objeto), Amazon S3 evalúa las políticas de recursos correspondientes (política de bucket, ACL de bucket y ACL de objeto) al mismo tiempo.

> **Note**
>
> Si la Cuenta de AWS principal es propietaria del recurso (bucket u objeto), puede conceder permisos de recursos a su entidad principal de IAM mediante la política de usuario o la política de recursos.

2. Contexto de bucket: en este contexto, Amazon S3 evalúa las políticas pertenecientes a la Cuenta de AWS que es propietaria del bucket.

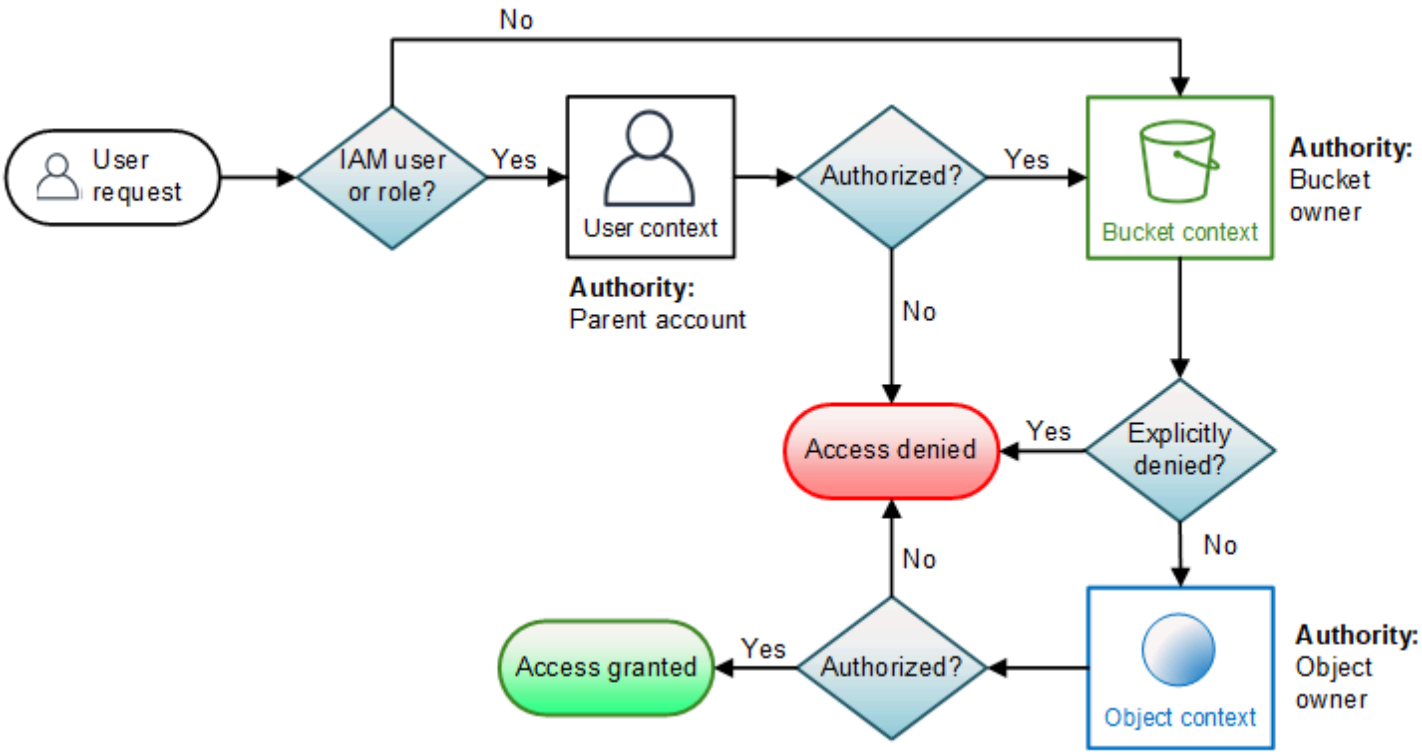
   Si la Cuenta de AWS que es propietaria del objeto en la solicitud no es la misma que la propietaria del bucket, Amazon S3 verifica las políticas para determinar si el propietario del bucket denegó explícitamente el acceso al objeto. Si se estableció una denegación explícita en el objeto, Amazon S3 no autoriza la solicitud.

3. Contexto de objeto: el solicitante debe tener permisos del propietario del objeto para realizar una operación de objeto específica. En este paso, Amazon S3 evalúa la ACL de objeto.

> **Note**
>
> Si los propietarios del bucket y del objeto son los mismos, el acceso al objeto se puede conceder en la política de bucket, que se evalúa en el contexto de bucket. Si los propietarios son diferentes, los propietarios del objeto deben utilizar una ACL de objeto para conceder permisos. Si la Cuenta de AWS que es propietaria del objeto es también la cuenta principal a la que pertenece la entidad principal de IAM, esta puede configurar los permisos del objeto en una política de usuario, que se evalúa en el contexto de usuario. Para obtener más información acera del uso de estas alternativas de políticas de acceso, consulte Explicaciones que utilizan políticas para administrar el acceso a los recursos de Amazon S3.
>
> Si, como propietario del bucket, desea tener la propiedad de todos los objetos del bucket y utilizar políticas de bucket o políticas basadas en IAM para administrar el acceso a estos objetos, puede aplicar la configuración de propietario del bucket obligatorio de Object Ownership. Con esta configuración, como propietario del bucket, tiene automáticamente la propiedad y el control total sobre cada objeto del bucket. Las ACL de bucket y objeto

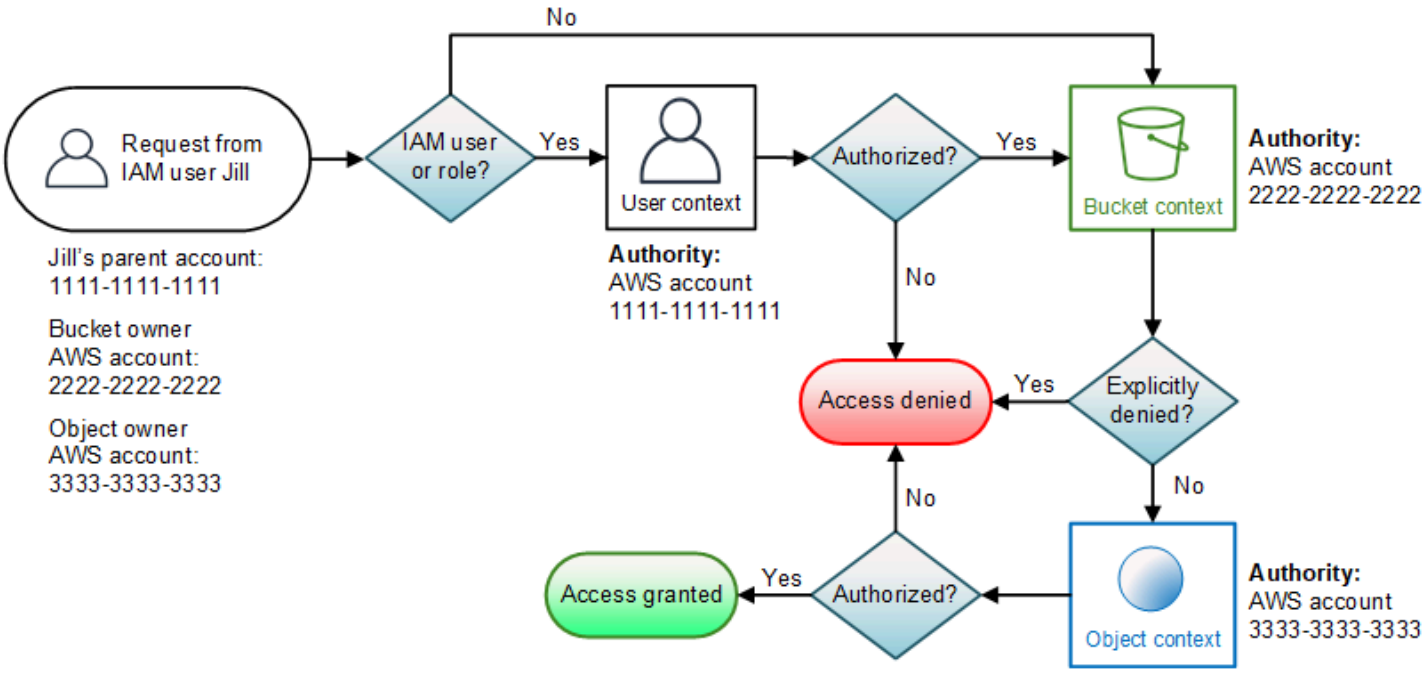
no se pueden editar y ya no se consideran para el acceso. Para obtener más información, consulte Control de la propiedad de los objetos y desactivación de las ACL del bucket.

A continuación se muestra una ilustración de la evaluación de una operación de objeto basada en el contexto.



Ejemplo de una solicitud de operación de objeto

En este ejemplo, Jill, la usuaria de IAM cuya Cuenta de AWS principal es 1111-1111-1111, envía una solicitud de operación de objeto (por ejemplo: `GetObject`) para un objeto que pertenece a la Cuenta de AWS 3333-3333-3333 en un bucket propiedad de la Cuenta de AWS 2222-2222-2222.



Jill necesitará contar con los permisos correspondientes de la Cuenta de AWS principal, el propietario del bucket y el propietario del objeto. Amazon S3 evalúa el contexto de la siguiente manera:

1. Dado que la solicitud es de una entidad principal de IAM, Amazon S3 evalúa el contexto de usuario para verificar que la Cuenta de AWS principal 1111-1111-1111 haya otorgado a Jill el permiso necesario para realizar la operación solicitada. Si Jill tiene permiso, Amazon S3 evalúa el contexto de bucket. De lo contrario, Amazon S3 deniega la solicitud.
2. En el contexto del bucket, el propietario del bucket, la Cuenta de AWS 2222-2222-2222, es la autoridad del contexto. Amazon S3 evalúa la política de bucket para determinar si el propietario del bucket le denegó explícitamente el acceso al objeto a Jill.
3. En el contexto de objeto, la autoridad del contexto es la Cuenta de AWS 3333-3333-3333, propietaria del objeto. Amazon S3 evalúa la ACL de objeto para determinar si Jill tiene permiso para acceder al objeto. Si lo tiene, Amazon S3 autoriza la solicitud.

Políticas administradas por AWS para Amazon S3

Una política administrada de AWS es una política independiente que AWS crea y administra. Las políticas administradas de AWS se diseñan para ofrecer permisos para muchos casos de uso comunes, por lo que puede empezar a asignar permisos a los usuarios, grupos y roles.

Considere que es posible que las políticas administradas de AWS no concedan permisos de privilegio mínimo para los casos de uso concretos, ya que están disponibles para que las utilicen todos los clientes de AWS. Se recomienda definir políticas administradas por el cliente específicas para sus casos de uso a fin de reducir aún más los permisos.

No puede cambiar los permisos definidos en las políticas administradas de AWS. Si AWS actualiza los permisos definidos en una política administrada de AWS, la actualización afecta a todas las identidades de entidades principales (usuarios, grupos y roles) a las que está adjunta la política. Lo más probable es que AWS actualice una política administrada de AWS cuando se lance un nuevo Servicio de AWS o las operaciones de la API nuevas estén disponibles para los servicios existentes.

Para obtener más información, consulte Políticas administradas por AWS en la Guía del usuario de IAM.

Política administrada de AWS: AmazonS3FullAccess

Puede adjuntar la política `AmazonS3FullAccess` a las identidades de IAM. Esta política otorga permisos que brindan acceso completo a Amazon S3.

Para ver los permisos de esta política, consulte AmazonS3FullAccess en la AWS Management Console.

Política administrada de AWS: AmazonS3ReadOnlyAccess

Puede adjuntar la política `AmazonS3ReadOnlyAccess` a las identidades de IAM. Esta política otorga permisos que brindan acceso de solo lectura a Amazon S3.

Para ver los permisos de esta política, consulte AmazonS3ReadOnlyAccess en la AWS Management Console.

Política administrada de AWS: AmazonS3ObjectLambdaExecutionRolePolicy

Proporciona a las funciones de AWS Lambda los permisos necesarios para enviar datos a S3 Object Lambda cuando se realizan solicitudes a un punto de acceso S3 Object Lambda. También concede permisos a Lambda para escribir en Amazon CloudWatch Logs.

Para ver los permisos de esta política, consulte AmazonS3ObjectLambdaExecutionRolePolicy en la AWS Management Console.

Actualizaciones de Amazon S3 en las políticas administradas por AWS

Es posible consultar los detalles sobre las actualizaciones de las políticas administradas por AWS para Amazon S3 debido a que este servicio comenzó a realizar el seguimiento de estos cambios.

Cambio	Descripción	Fecha
Amazon S3 agregó permisos Describe a `AmazonS3ReadOnlyAccess`	Amazon S3 agregó permisos `s3:Describe*` a `AmazonS3ReadOnlyAccess` .	11 de agosto de 2023
Amazon S3 agregó permisos de S3 Object Lambda a `AmazonS3FullAccess` y `AmazonS3ReadOnlyAccess`	Amazon S3 actualizó el `AmazonS3FullAccess` y las políticas de `AmazonS3ReadOnlyAccess` para incluir permisos para S3 Object Lambda.	27 de septiembre de 2021
Amazon S3 agregó `AmazonS3ObjectLambdaExecutionRolePolicy`	Amazon S3 ha agregado una nueva política administrada por AWS llamada `AmazonS3ObjectLambdaExecutionRolePolicy` que proporciona permisos a las funciones de Lambda para interactuar con S3 Object Lambda y escribir en CloudWatch Logs.	18 de agosto de 2021
Amazon S3 comenzó a realizar el seguimiento de los cambios	Amazon S3 comenzó a realizar el seguimiento de los cambios en sus políticas administradas de AWS	18 de agosto de 2021

Uso de roles vinculados a servicios para Amazon S3 Storage Lens

Si desea utilizar Amazon S3 Storage Lens para recopilar y agrupar métricas en todas sus cuentas de AWS Organizations, primero debe asegurarse de que S3 Storage Lens tenga el acceso de confianza habilitado por la cuenta de administración de la organización. Lente de almacenamiento de S3 crea un rol vinculado a servicios (SLR) para permitirle que obtenga la lista de Cuentas de AWS que pertenecen a la organización. Esta lista de cuentas la utiliza S3 Storage Lens para recopilar métricas de los recursos de S3 en todas las cuentas de miembros cuando se crean o actualizan el panel o las configuraciones de S3 Storage Lens.

Amazon S3 Storage Lens utiliza roles vinculados a servicios de AWS Identity and Access Management (IAM). Un rol vinculado a un servicio es un tipo único de rol de IAM que está vinculado directamente a S3 Storage Lens. Los roles vinculados a servicios están predefinidos por Lente de almacenamiento de S3 e incluyen todos los permisos que el servicio requiere para llamar a otros Servicios de AWS en su nombre.

Un rol vinculado a un servicio simplifica la configuración de S3 Storage Lens porque ya no tendrá que agregar manualmente los permisos necesarios. S3 Storage Lens define los permisos de sus roles vinculados a servicios y, a menos que esté definido de otra manera, solo S3 Storage Lens puede asumir sus roles. Los permisos definidos incluyen las políticas de confianza y de permisos, y que la política de permisos no se puede adjuntar a ninguna otra entidad de IAM.

Solo puede eliminar este rol vinculado a servicios después de eliminar los recursos relacionados. De esta forma, se protegen los recursos de S3 Storage Lens, ya que evita que se puedan eliminar accidentalmente permisos de acceso a los recursos.

Para obtener información acerca de otros servicios que admiten roles vinculados a servicios, consulte Servicios de AWS que funcionan con IAM y busque los servicios que muestran Yes (Sí) en la columna Service Linked Role (Rol vinculado a servicios). Seleccione una opción Sí con un enlace para ver la documentación acerca del rol vinculado al servicio en cuestión.

Permisos de roles vinculados a servicios para Amazon S3 Storage Lens

S3 Storage Lens utiliza el rol vinculado a servicios denominado AWSServiceRoleForS3StorageLens. Esto permite el acceso a los servicios y los recursos de AWS utilizados o administrados por S3 Storage Lens. Permite que S3 Storage Lens acceda a los recursos de AWS Organizations en su nombre.

El rol vinculado al servicio de S3 Storage Lens confía en el siguiente servicio en el almacenamiento de la organización:

- `storage-lens.s3.amazonaws.com`

La política de permisos del rol permite que S3 Storage Lens realice las siguientes acciones en los recursos:

- `organizations:DescribeOrganization`

  `organizations:ListAccounts`

  `organizations:ListAWSServiceAccessForOrganization`

  `organizations:ListDelegatedAdministrators`

Debe configurar permisos para permitir a una entidad de IAM (como un usuario, grupo o función) crear, editar o eliminar la descripción de una función vinculada a un servicio. Para obtener más información, consulte Permisos de roles vinculados a servicios en la Guía del usuario de IAM.

Creación de un rol vinculado a un servicio para S3 Storage Lens

No necesita crear manualmente un rol vinculado a un servicio. Cuando completa alguna de las siguientes tareas habiendo iniciado sesión en las cuentas de administración o de administrador delegado de AWS Organizations, S3 Storage Lens crea el rol vinculado a servicios por usted:

- Cree una configuración del panel de S3 Storage Lens para la organización en la consola de Amazon S3.
- Realice una operación PUT de una configuración de S3 Storage Lens para la organización mediante la API de REST, la AWS CLI y los SDK.

> **Note**
>
> Lente de almacenamiento de S3 admitirá un máximo de cinco administradores delegados por organización.

Si elimina este rol vinculado a servicios, las acciones anteriores lo volverán a crear según sea necesario.

Ejemplo de política para el rol vinculado al servicio de S3 Storage Lens

Example Política de permisos para el rol vinculado al servicio de S3 Storage Lens

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Sid": "AwsOrgsAccess",
            "Effect": "Allow",
            "Action": [
                "organizations:DescribeOrganization",
                "organizations:ListAccounts",
                "organizations:ListAWSServiceAccessForOrganization",
                "organizations:ListDelegatedAdministrators"
            ],
            "Resource": [
                "*"
            ]
        }
    ]
}
```

Edición de un rol vinculado a servicios para Amazon S3 Storage Lens

Lente de almacenamiento de S3 no le permite editar el rol vinculado al servicio AWSServiceRoleForS3StorageLens. Después de crear un rol vinculado a un servicio, no puede cambiarle el nombre, ya que varias entidades pueden hacer referencia a él. Sin embargo, puede editar la descripción del rol mediante IAM. Para obtener más información, consulte Editar un rol vinculado a servicios en la Guía del usuario de IAM.

Eliminación de un rol vinculado a servicios para Amazon S3 Storage Lens

Si ya no utiliza el rol vinculado a servicios, le recomendamos que elimine dicho rol. De esta forma no tiene una entidad no utilizada que no se monitoree ni mantenga de forma activa. Sin embargo, debe limpiar los recursos del rol vinculado al servicio antes de eliminarlo manualmente.

> **Note**
>
> Si el servicio de Amazon S3 Storage Lens utiliza el rol cuando intenta eliminar los recursos, es posible que no se pueda borrar. En tal caso, espere unos minutos e intente de nuevo la operación.

Para eliminar AWSServiceRoleForS3StorageLens, debe eliminar todas las configuraciones de Lente de almacenamiento de S3 de la organización presentes en todas las Regiones de AWS mediante las cuentas de administración o de administrador delegado de AWS Organizations.

Los recursos son configuraciones de S3 Storage Lens de organización. Utilice Lente de almacenamiento de S3 para limpiar los recursos y, a continuación, utilice la consola de IAM, la CLI, la API de REST o el AWS SDK para eliminar el rol.

En la API de REST, la AWS CLI y los SDK, se pueden descubrir configuraciones de Lente de almacenamiento de S3 con `ListStorageLensConfigurations` en todas las regiones en las que la organización haya creado configuraciones de Lente de almacenamiento de S3. Utilice la acción `DeleteStorageLensConfiguration` para eliminar estas configuraciones de modo que pueda eliminar el rol.

> **Note**
>
> Para eliminar el rol vinculado a servicios, debe eliminar todas las configuraciones de S3 Storage Lens de la organización en todas las regiones donde existan.

Eliminación de los recursos de Lente de almacenamiento de Amazon S3 utilizados por el SLR AWSServiceRoleForS3StorageLens

1. Para obtener una lista de configuraciones en el nivel de la organización, debe usar `ListStorageLensConfigurations` en cada región que tiene configuraciones de Lente de almacenamiento de S3. Esta lista también se puede obtener en la consola de Amazon S3.
2. Elimine estas configuraciones de los puntos de conexión regionales apropiados al invocar la llamada a la API `DeleteStorageLensConfiguration` o a través de la consola de Amazon S3.

Eliminación manual del rol vinculado a servicios mediante IAM

Una vez eliminadas las configuraciones, elimine el SLR AWSServiceRoleForS3StorageLens de la consola de IAM o invoque la API de IAM `DeleteServiceLinkedRole` o use la AWS CLI o el AWS SDK. Para obtener más información, consulte Eliminación de un rol vinculado a servicios en la Guía del usuario de IAM.

Regiones admitidas para roles vinculados a servicios de S3 Storage Lens

S3 Storage Lens admite el uso de roles vinculados a servicios en todas las Regiones de AWS en las que el servicio esté disponible. Para obtener más información, consulte Regiones y puntos de enlace de Amazon S3 Storage Lens.

Solución de problemas de identidad y acceso de Amazon S3

Utilice la siguiente información para diagnosticar y solucionar los problemas comunes que es posible que surjan cuando se trabaja con Amazon S3 e IAM.

Temas

- He recibido un error de acceso denegado
- No tengo autorización para realizar una acción en Amazon S3
- No tengo autorización para realizar la operación iam:PassRole
- Quiero permitir a personas externas a mi Cuenta de AWS el acceso a mis recursos de Amazon S3

He recibido un error de acceso denegado

Verifique que no hay ninguna instrucción `Deny` explícita para el solicitante al que intenta conceder permisos, ni en la política de bucket ni en la política basada en la identidad.

Para obtener información detallada sobre la solución de errores de acceso denegado, consulte Solucionar errores de acceso denegado (403 Prohibido) en Amazon S3.

No tengo autorización para realizar una acción en Amazon S3

Si recibe un error que indica que no tiene autorización para realizar una acción, las políticas se deben actualizar para permitirle realizar la acción.

En el siguiente ejemplo, el error se produce cuando el usuario de IAM `mateojackson` intenta utilizar la consola para consultar los detalles acerca de un recurso ficticio *`my-example-widget`*, pero no tiene los permisos ficticios `s3:`*`GetWidget`*.

```
User: arn:aws:iam::123456789012:user/mateojackson is not authorized to perform:
 s3:GetWidget on resource: my-example-widget
```

En este caso, la política del usuario `mateojackson` debe actualizarse para permitir el acceso al recurso *`my-example-widget`* mediante la acción `s3:GetWidget`.

Si necesita ayuda, póngase en contacto con su administrador de AWS. El administrador es la persona que le proporcionó las credenciales de inicio de sesión.

No tengo autorización para realizar la operación iam:PassRole

Si recibe un error que indica que no tiene autorización para realizar la acción `iam:PassRole`, las políticas se deben actualizar para permitirle pasar un rol a Amazon S3.

Algunos servicios de Servicios de AWS le permiten transferir un rol existente a dicho servicio en lugar de crear un nuevo rol de servicio o uno vinculado al servicio. Para ello, debe tener permisos para transferir el rol al servicio.

En el siguiente ejemplo, el error se produce cuando un usuario de IAM denominado `marymajor` intenta utilizar la consola para realizar una acción en Amazon S3. Sin embargo, la acción requiere que el servicio cuente con permisos que otorguen un rol de servicio. Mary no tiene permisos para transferir el rol al servicio.

```
User: arn:aws:iam::123456789012:user/marymajor is not authorized to perform:
 iam:PassRole
```

En este caso, las políticas de Mary se deben actualizar para permitirle realizar la acción `iam:PassRole`.

Si necesita ayuda, póngase en contacto con su administrador de AWS. El administrador es la persona que le proporcionó las credenciales de inicio de sesión.

Quiero permitir a personas externas a mi Cuenta de AWS el acceso a mis recursos de Amazon S3

Puede crear un rol que los usuarios de otras cuentas o las personas externas a la organización puedan utilizar para acceder a sus recursos. Puede especificar una persona de confianza para que asuma el rol. En el caso de los servicios que admitan las políticas basadas en recursos o las listas de control de acceso (ACL), puede utilizar dichas políticas para conceder a las personas acceso a sus recursos.

Para más información, consulte lo siguiente:

- Para saber si Amazon S3 admite estas características, consulte Cómo funciona Amazon S3 con IAM.
- Para obtener información acerca de cómo proporcionar acceso a los recursos de las Cuenta de AWS de su propiedad, consulte Proporcionar acceso a un usuario de IAM a otra Cuentas de AWS de la que es propietario en la Guía del usuario de IAM.
- Para obtener información acerca de cómo proporcionar acceso a tus recursos a Cuentas de AWS de terceros, consulte Proporcionar acceso a Cuentas de AWS que son propiedad de terceros en la Guía del usuario de IAM.
- Para obtener información sobre cómo proporcionar acceso mediante una federación de identidades, consulte Proporcionar acceso a usuarios autenticados externamente (identidad federada) en la Guía del usuario de IAM.
- Para conocer sobre la diferencia entre los roles y las políticas basadas en recursos para el acceso entre cuentas, consulte Cross account resource access in IAM en la Guía del usuario de IAM.

Administración del acceso con S3 Access Grants

Para cumplir con el principio de privilegio mínimo, debe definir el acceso granular a sus datos de Amazon S3 en función de las aplicaciones, las personas, los grupos o las unidades organizativas. Según la escala y la complejidad de los patrones de acceso, puede utilizar varios enfoques para lograr un acceso granular a sus datos en Amazon S3.

El enfoque más sencillo para administrar el acceso a números pequeños a medianos de conjuntos de datos en Amazon S3 por parte de las entidades principales de AWS Identity and Access Management (IAM) consiste en definir las políticas de permisos de IAM y las políticas de buckets de S3. Esta estrategia funciona siempre que las políticas necesarias se ajusten a los límites de tamaño de las políticas de buckets de S3 (20 KB) y de las políticas de IAM (5 KB) y al número de entidades principales de IAM permitidas por cuenta.

A medida que escale el número de conjuntos de datos y casos de uso, es posible que necesite más espacio para las políticas. Un enfoque que ofrece mucho más espacio para instrucciones de política consiste en utilizar los puntos de acceso de S3 como puntos de conexión adicionales para los buckets de S3, ya que cada punto de acceso puede tener su propia política. Puede definir patrones de control de acceso bastante granulares, ya que puede tener miles de puntos de acceso por Región de AWS por cuenta, con una política de hasta 20 KB de tamaño para cada punto de acceso. Aunque

los puntos de acceso de S3 aumentan la cantidad de espacio disponible para las políticas, requieren un mecanismo para que los clientes descubran el punto de acceso correcto para el conjunto de datos adecuado.

Un tercer enfoque consiste en implementar un patrón de agente de sesiones de IAM, en el que se implementa una lógica de decisión de acceso y se generan dinámicamente credenciales de sesión de IAM a corto plazo para cada sesión de acceso. Aunque el enfoque de agente de sesiones de IAM admite patrones de permisos dinámicos arbitrarios y se escala de forma eficaz, es necesario desarrollar la lógica del patrón de acceso.

En lugar de usar estos enfoques, puede utilizar S3 Access Grants para administrar el acceso a sus datos de Amazon S3. S3 Access Grants proporciona un modelo simplificado para definir los permisos de acceso a los datos en Amazon S3 por prefijo, bucket u objeto. Además, puede utilizar S3 Access Grants para conceder acceso tanto a las entidades principales de IAM como directamente a los usuarios o grupos de su directorio corporativo.

Por lo general, los permisos para los datos de Amazon S3 se definen asignando usuarios y grupos a conjuntos de datos. Puede usar S3 Access Grants para definir las asignaciones de acceso directo de los prefijos de S3 a los usuarios y roles dentro de los buckets y objetos de Amazon S3. Con el esquema de acceso simplificado de S3 Access Grants, puede conceder acceso de solo lectura, de solo escritura o de lectura y escritura por prefijo de S3 tanto a las entidades principales de IAM como directamente a los usuarios o grupos de un directorio corporativo. Con estas funciones de S3 Access Grants, las aplicaciones pueden solicitar datos de Amazon S3 en nombre del usuario autenticado actual de la aplicación.

Al integrar S3 Access Grants con la característica de propagación de identidad de confianza de AWS IAM Identity Center, sus aplicaciones pueden realizar solicitudes a Servicios de AWS (incluido S3 Access Grants) directamente en nombre de un usuario autenticado del directorio corporativo. Ya no es necesario que sus aplicaciones asignen primero al usuario a una entidad principal de IAM. Además, puesto que las identidades de los usuarios finales se propagan hasta Amazon S3, se simplifica la auditoría de qué usuario ha accedido a qué objeto de S3. Ya no es necesario reconstruir la relación entre los distintos usuarios y las sesiones de IAM. Cuando utiliza S3 Access Grants con la propagación de identidad de confianza del Centro de identidades de IAM, cada evento de datos de AWS CloudTrail de Amazon S3 contiene una referencia directa al usuario final en cuyo nombre se accedió a los datos.

Consulte los siguientes temas para obtener más información acerca de las S3 Access Grants.

Temas

Conceptos de S3 Access Grants

Flujo de trabajo de Concesiones de acceso a Amazon S3

El flujo de trabajo de Concesiones de acceso a Amazon S3 es el siguiente:

1. Cree una instancia de Concesiones de acceso a Amazon S3. Consulte Crear una instancia de S3 Access Grants.
2. En su instancia de Concesiones de acceso a Amazon S3, registre las ubicaciones en sus datos de Amazon S3 y asigne estas ubicaciones a roles de AWS Identity and Access Management (IAM). Consulte Registrar una ubicación.
3. Cree concesiones para beneficiarios, que les den acceso a estos a sus recursos de S3. Consulte Crear concesiones.
4. El beneficiario solicita credenciales temporales de Concesiones de acceso a Amazon S3. Consulte Solicitar acceso a los datos de Amazon S3 a través de S3 Access Grants.
5. El concesionario accede a los datos de S3 con esas credenciales temporales. Consulte Acceder a los datos de S3 mediante una concesión de acceso.

Para obtener más información, consulte Introducción a S3 Access Grants.

Instancias de S3 Access Grants

Una instancia de Concesiones de acceso a Amazon S3 es un contenedor lógico para concesiones individuales. Al crear una instancia de Concesiones de acceso a Amazon S3, debe especificar una Región de AWS. Cada Región de AWS de su Cuenta de AWS puede tener una instancia de Concesiones de acceso a Amazon S3. Para obtener más información, consulte Crear una instancia de S3 Access Grants.

Si quiere usar Concesiones de acceso a Amazon S3 para conceder acceso a identidades de usuarios y grupos de su directorio corporativo, también debe asociar su instancia de Concesiones de acceso a Amazon S3 a una instancia de AWS IAM Identity Center. Para obtener más información, consulte S3 Access Grants e identidades de directorios corporativos.

Una instancia de Concesiones de acceso a Amazon S3 recién creada está vacía. Debe registrar una ubicación en la instancia, que puede ser la ruta predeterminada de S3 (`s3://`), un bucket o un prefijo dentro de un bucket. Después de registrar al menos una ubicación, puede crear concesiones de acceso que concedan acceso a los datos de esta ubicación registrada.

Ubicaciones

Una ubicación de Concesiones de acceso a Amazon S3 asigna buckets o prefijos a un rol de AWS Identity and Access Management (IAM). Concesiones de acceso a Amazon S3 asume este rol de IAM para vender credenciales temporales al beneficiario que accede a esa ubicación en particular. En primer lugar, debe registrar al menos una ubicación en su instancia de Concesiones de acceso a Amazon S3 para poder crear una concesión de acceso.

Le recomendamos que registre la ubicación predeterminada (`s3://`) y la asigne a un rol de IAM. La ubicación en la ruta de S3 predeterminada (`s3://`) cubre el acceso a todos los buckets de S3 en la Región de AWS de su cuenta. Al crear una concesión de acceso, puede limitar el ámbito de la concesión a un bucket, un prefijo o un objeto dentro de la ubicación predeterminada.

Los casos de uso de administración de acceso más complejos pueden requerir que registre una ubicación diferente a la predeterminada. Estos son algunos ejemplos de estos casos de uso:

- Supongamos que *`amzn-s3-demo-bucket`* es una ubicación registrada en su instancia de Concesiones de acceso a Amazon S3 con un rol de IAM asignado, pero a este rol de IAM se le niega el acceso a un prefijo concreto del bucket. En este caso, puede registrar el prefijo al que el rol de IAM no tiene acceso como una ubicación independiente y asignar esa ubicación a un rol de IAM diferente con el acceso necesario.
- Supongamos que desea crear concesiones que restrinjan el acceso solo a los usuarios dentro de un punto de conexión a la nube privada virtual (VPC). En este caso, puede registrar una

ubicación para un bucket en el que el rol de IAM restrinja el acceso al punto de conexión de VPC. Más adelante, cuando un beneficiario solicite las credenciales a Concesiones de acceso a Amazon S3, este asumirá el rol de IAM de la ubicación para vender las credenciales temporales. Esta credencial denegará el acceso al bucket específico, a menos que el iniciador se encuentre dentro del punto de conexión de VPC. Esta denegación de permiso se aplica además del permiso normal READ, WRITE o READWRITE que se especifica en la concesión.

Si su caso de uso requiere que registre varias ubicaciones en su instancia de Concesiones de acceso a Amazon S3, puede registrar cualquiera de las siguientes opciones:

- La ubicación predeterminada de S3 (`s3://`).
- Un bucket (por ejemplo, *`amzn-s3-demo-bucket`*) o varios buckets
- Un bucket y un prefijo (por ejemplo, *`amzn-s3-demo-bucket`*`/prefix*`) o varios prefijos

Para obtener información sobre el número máximo de ubicaciones que puede registrar en su instancia de Concesiones de acceso a Amazon S3, consulte Limitaciones de S3 Access Grants. Para obtener más información sobre el registro de una ubicación de Concesiones de acceso a Amazon S3, consulte Registrar una ubicación.

Tras registrar la primera ubicación en su instancia de Concesiones de acceso a Amazon S3, la instancia seguirá sin tener ninguna concesión de acceso individual. Por lo tanto, aún no se ha concedido acceso a ninguno de sus datos de S3. Ahora puede crear concesiones de acceso para conceder acceso. Para obtener más información acerca de la creación de concesiones, consulte Crear concesiones.

Concesiones

Una concesión individual en una instancia de Concesiones de acceso a Amazon S3 permite que una identidad específica (una entidad principal de IAM o un usuario o grupo de un directorio corporativo) acceda a una ubicación que esté registrada en su instancia de Concesiones de acceso a Amazon S3.

Al crear una concesión, no es necesario conceder acceso a toda la ubicación registrada. Puede limitar el ámbito de acceso de la concesión en una ubicación. Si la ubicación registrada es la ruta S3 predeterminada (`s3://`), debe limitar el ámbito de la concesión a un bucket, a un prefijo dentro de un bucket o a un objeto específico. Si la ubicación registrada de la concesión es un bucket o un prefijo, puede dar acceso a todo el bucket o prefijo o, si lo prefiere, puede limitar el ámbito de la concesión a un prefijo, subprefijo u objeto.

En la concesión, también establece el nivel de acceso de la concesión a READ, WRITE o READWRITE. Supongamos que tiene una concesión que permite al grupo del directorio corporativo `01234567-89ab-cdef-0123-456789abcdef` acceso READ al bucket `s3://amzn-s3-demo-bucket/projects/items/*`. Los usuarios de este grupo pueden tener acceso READ a todos los objetos que tengan un nombre de clave de objeto que comience con el prefijo `projects/items/` en el bucket denominado *amzn-s3-demo-bucket*.

Para obtener información sobre el número máximo de concesiones que puede crear en su instancia de Concesiones de acceso a Amazon S3, consulte Limitaciones de S3 Access Grants. Para obtener más información acerca de la creación de concesiones, consulte Crear concesiones.

Credenciales temporales de S3 Access Grants

Tras crear una concesión, una aplicación autorizada que utilice la identidad especificada en la concesión podrá solicitar credenciales de acceso justo a tiempo. Para ello, la aplicación llama a la operación de API de S3 GetDataAccess. Los beneficiarios pueden usar esta operación de API para solicitar acceso a los datos de S3 que ha compartido con ellos.

La instancia de S3 Access Grants evalúa la solución `GetDataAccess` con respecto a las concesiones de las que dispone. Si existe una concesión correspondiente para el solicitante, Concesiones de acceso a Amazon S3 asume el rol de IAM asociado a la ubicación registrada de la concesión correspondiente. Concesiones de acceso a Amazon S3 concede los permisos de las credenciales temporales para acceder únicamente al bucket, prefijo u objeto de S3 especificado en el ámbito de la concesión.

El tiempo de vencimiento de las credenciales de acceso temporal es de 1 hora de forma predeterminada, pero puede establecerlo en cualquier valor, desde 15 minutos a 12 horas. Consulte la duración máxima de la sesión en la referencia de la API AssumeRole.

Funcionamiento

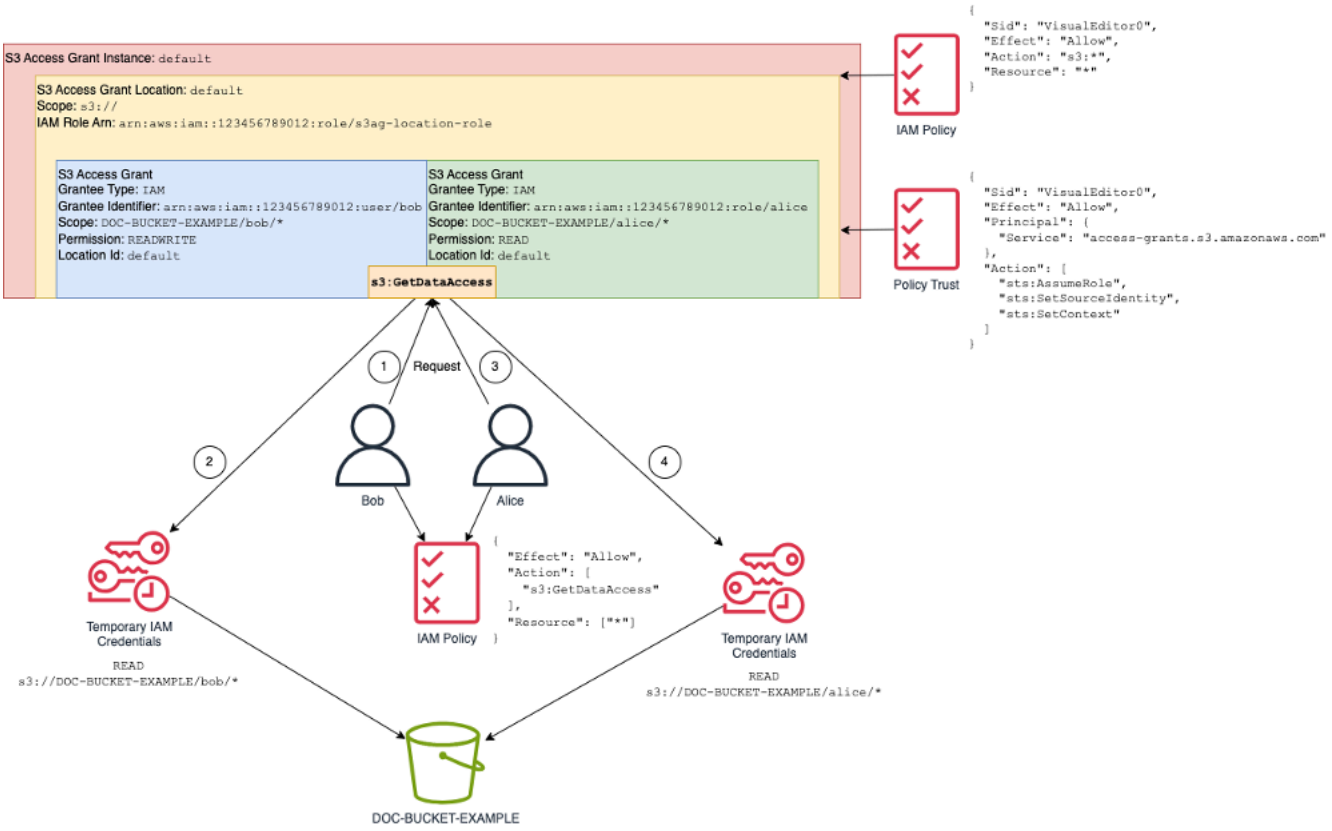
En el siguiente diagrama, una ubicación predeterminada de Amazon S3 con el alcance `s3://` está registrada con el rol de IAM `s3ag-location-role`. Este rol de IAM tiene permisos para realizar acciones de Amazon S3 dentro de la cuenta cuando sus credenciales se obtienen a través de S3 Access Grants.

En esta ubicación, se crean dos concesiones de acceso individuales para dos usuarios de IAM. Al usuario de IAM, Bob, se le concede acceso de `READ` y `WRITE` con el prefijo `bob/` del bucket

DOC-BUCKET-EXAMPLE. A otro rol de IAM, Alice, solo se le concede acceso de READ con el prefijo alice/ del bucket DOC-BUCKET-EXAMPLE. Se define una concesión, coloreada en azul, para que Bob acceda al prefijo bob/ del bucket DOC-BUCKET-EXAMPLE. Se define una concesión, coloreada en verde, para que Alice acceda al prefijo del alice/ del bucket DOC-BUCKET-EXAMPLE.

Cuando Bob READ los datos, el rol de IAM asociado a la ubicación en la que se encuentra su concesión llama a la operación de la API GetDataAccess de S3 Access Grants. Si Bob intenta READ cualquier prefijo u objeto de S3 que comience por s3://DOC-BUCKET-EXAMPLE/bob/*, la solicitud GetDataAccess devuelve un conjunto de credenciales de sesión de IAM temporales con permiso para s3://DOC-BUCKET-EXAMPLE/bob/*. Del mismo modo, Bob puede WRITE en cualquier prefijo u objeto de S3 que comience por s3://DOC-BUCKET-EXAMPLE/bob/*, ya que la concesión también lo permite.

Del mismo modo, Alice puede READ cualquier cosa que comience con s3://DOC-BUCKET-EXAMPLE/alice/. Sin embargo, si intenta WRITE cualquier cosa con cualquier bucket, prefijo u objeto en s3://, recibirá un mensaje de error de acceso denegado (403 Prohibido), ya que no hay ninguna concesión que le brinde acceso de WRITE a ningún dato. Además, si Alice solicita algún nivel de acceso (READ o WRITE) a datos fuera de s3://DOC-BUCKET-EXAMPLE/alice/, volverá a recibir un error de acceso denegado.



Este patrón se adapta a un gran número de usuarios y buckets, y simplifica la administración de esos permisos. En lugar de editar políticas de bucket de S3 potencialmente voluminosas cada vez que quiera añadir o eliminar una relación de acceso a un prefijo de usuario individual, puede añadir y eliminar concesiones individuales y discretas.

S3 Access Grants e identidades de directorios corporativos

Puede usar Amazon S3 Access Grants para otorgar acceso a las entidades principales de AWS Identity and Access Management (IAM) (usuarios o roles), tanto en la misma Cuenta de AWS como en otras. Sin embargo, en muchos casos, la entidad que accede a los datos es un usuario final de su directorio corporativo. En lugar de otorgar acceso a las entidades principales de IAM, puede utilizar Amazon S3 Access Grants para otorgar acceso directamente a los usuarios y grupos corporativos. Con Amazon S3 Access Grants, ya no es necesario asignar las identidades corporativas a las entidades principales de IAM intermedias para poder acceder a los datos de S3 mediante sus aplicaciones corporativas.

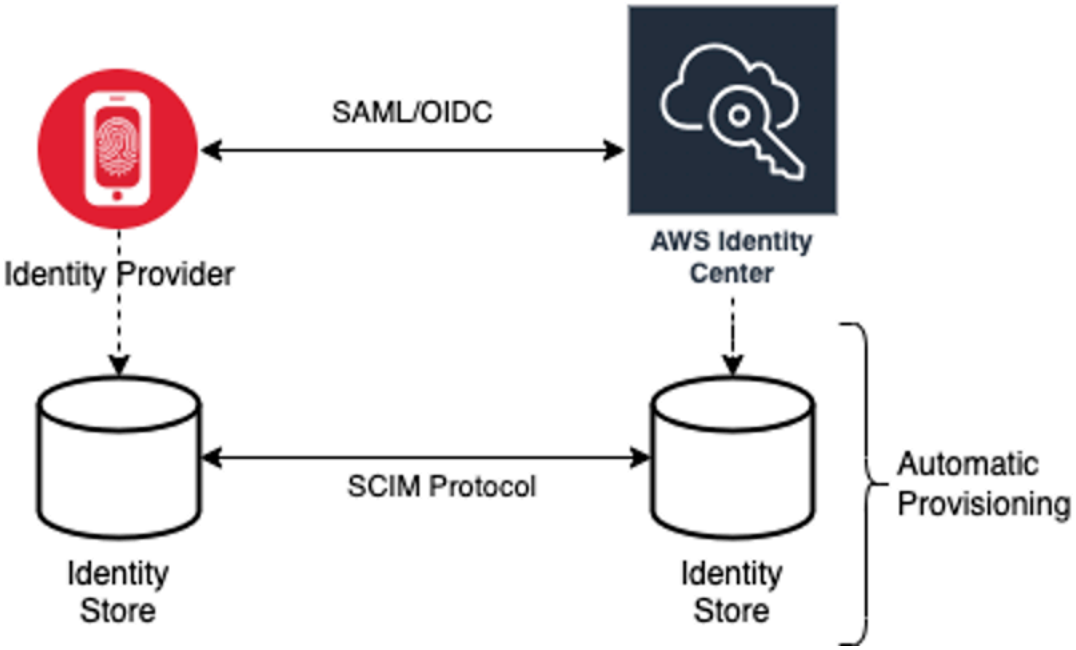
Esta nueva funcionalidad, que permite utilizar las identidades de los usuarios finales para acceder a los datos, se proporciona al asociar su instancia de S3 Access Grants a una instancia. del AWS IAM Identity Center. El Centro de identidades de IAM es compatible con proveedores de identidades basados en estándares y es el centro de AWS para todos los servicios o características, incluido S3 Access Grants, que admiten identidades de usuarios finales. El Centro de identidades de IAM proporciona soporte de autenticación para identidades corporativas mediante su característica de propagación de identidades de confianza. Para obtener más información, consulte Propagación de identidades de confianza en aplicaciones.

Para empezar a utilizar el soporte de identidades de personal en S3 Access Grants, como requisito previo, comience en el Centro de identidades de IAM configurando el aprovisionamiento de identidades entre su proveedor de identidad corporativa y el Centro de identidades de IAM. El Centro de identidades de IAM es compatible con proveedores de identidad corporativa como Okta, Microsoft Entra ID (anteriormente Azure Active Directory) o cualquier otro proveedor de identidades (IdP) externo que admita el protocolo System for Cross-Domain Identity Management (SCIM). Cuando conecta el Centro de identidades de IAM a su IdP y habilita el aprovisionamiento automático, los usuarios y grupos de su IdP se sincronizan en el almacén de identidades del Centro de identidades de IAM. Tras este paso, el Centro de identidades de IAM tiene su propia vista de sus usuarios y grupos, de modo que puede hacer referencia a ellos mediante otros Servicios de AWS y características, como S3 Access Grants. Para obtener más información sobre la configuración del aprovisionamiento automático de Centro de identidades de IAM, consulte Aprovisionamiento automático en la Guía del usuario de AWS IAM Identity Center.

El Centro de identidades de IAM está integrado con AWS Organizations, para que pueda administrar de forma centralizada los permisos de varias Cuentas de AWS sin tener que configurar cada una de ellas manualmente. En una organización típica, el administrador de identidades configura una instancia del Centro de identidades de IAM para toda la organización, como un único punto de sincronización de identidades. Esta instancia del Centro de identidades de IAM suele ejecutarse en una Cuenta de AWS dedicada de su organización. En esta configuración común, puede hacer referencia a las identidades de usuarios y grupos en S3 Access Grants desde cualquier Cuenta de AWS de la organización.

Sin embargo, si su administrador de AWS Organizations aún no ha configurado una instancia central del Centro de identidades de IAM, puede crear una local en la misma cuenta que su instancia de S3 Access Grants. Esta configuración es más común para los casos de uso de prueba de concepto o de desarrollo local. En todos los casos, la instancia del Centro de identidades de IAM debe estar en la misma Región de AWS que la instancia de S3 Access Grants a la que se asociará.

En el siguiente diagrama de una configuración del Centro de identidades de IAM con un IdP externo, el IdP se configura con SCIM para sincronizar el almacén de identidades del IdP al almacén de identidades del Centro de identidades de IAM.



Para usar las identidades de su directorio corporativo con S3 Access Grants, haga lo siguiente:

- Configure el aprovisionamiento automático en el Centro de identidades de IAM para sincronizar la información de usuarios y grupos de su IdP con el Centro de identidades de IAM.

- Configure su origen de identidad externa en el Centro de identidades de IAM como un emisor de tokens de confianza. Para obtener más información, consulte Propagación de identidad de confianza en aplicaciones en la Guía del usuario del AWS IAM Identity Center.
- Asocie su instancia de S3 Access Grants a su instancia del Centro de identidades de IAM. Puede hacerlo al crear su instancia de S3 Access Grants. Si ya ha creado su instancia de S3 Access Grants, consulte Asociar o desasociar su instancia del Centro de identidades de IAM.

Cómo pueden acceder las identidades de directorio a los datos de S3

Suponga que tiene usuarios del directorio corporativo que necesitan acceder a sus datos de S3 a través de una aplicación corporativa, por ejemplo, una aplicación de visualización de documentos, que está integrada con su IdP externo (por ejemplo, Okta) para autenticar a los usuarios. La autenticación del usuario en estas aplicaciones se suele realizar mediante redireccionamientos en el navegador web del usuario. Puesto que los usuarios del directorio no son las entidades principales de IAM, su aplicación necesita credenciales de IAM con las que pueda llamar a la operación de la API `GetDataAccess` de S3 Access Grants para obtener credenciales de acceso a los datos de S3 en nombre de los usuarios. A diferencia de los usuarios y roles de IAM, que obtienen las credenciales ellos mismos, su aplicación necesita una forma de representar a un usuario del directorio, que no esté asignado a un rol de IAM, para que el usuario pueda acceder a los datos a través de S3 Access Grants.

Esta transición, de un usuario de directorio autenticado a un intermediario de IAM que puede realizar solicitudes a S3 Access Grants en nombre del usuario del directorio, la realiza la aplicación a través de la característica del emisor de tokens de confianza del Centro de identidades de IAM. La aplicación, después de autenticar al usuario del directorio, tiene un token de identidad del IdP (por ejemplo, Okta) que representa al usuario del directorio según Okta. La configuración del emisor de tokens de confianza del Centro de identidades de IAM permite a la aplicación intercambiar este token de Okta (el inquilino de Okta está configurado como el "emisor de confianza") por un token de identidad diferente del Centro de identidades de IAM que representará de forma segura al usuario del directorio dentro de Servicios de AWS. La aplicación de datos asumirá entonces un rol de IAM y proporcionará el token del usuario del directorio del Centro de identidades de IAM como contexto adicional. La aplicación puede usar la sesión de IAM resultante para llamar a S3 Access Grants. El token representa tanto la identidad de la aplicación (la propia entidad principal de IAM) como la identidad del usuario del directorio.

El paso principal de esta transición es el intercambio de token. La aplicación realiza este intercambio de token mediante una llamada a la operación de la API `CreateTokenWithIAM` en el Centro de

identidades de IAM. Por supuesto, también se trata de una llamada a la API AWS y requiere que una entidad principal de IAM la firme. La entidad principal de IAM que realiza esta solicitud suele ser un rol de IAM asociado a la aplicación. Por ejemplo, si la aplicación se ejecuta en Amazon EC2, la solicitud `CreateTokenWithIAM` normalmente la realiza el rol de IAM asociado a la instancia EC2 en la que se ejecuta la aplicación. El resultado de una llamada `CreateTokenWithIAM` correcta es un nuevo token de identidad, que se reconocerá en Servicios de AWS.

El siguiente paso, antes de que la aplicación pueda llamar `GetDataAccess` en nombre del usuario del directorio, consiste en obtener una sesión de IAM que incluya la identidad del usuario del directorio. La aplicación lo hace con una solicitud `AssumeRole` de AWS Security Token Service (AWS STS) que también incluye el token del Centro de identidades de IAM para el usuario del directorio como contexto de identidad adicional. Este contexto adicional es lo que permite al Centro de identidades de IAM propagar la identidad del usuario del directorio para pasar al siguiente paso. El rol de IAM que asume la aplicación es el que necesitará los permisos de IAM para llamar a la operación `GetDataAccess`.

Tras haber asumido el rol de IAM portador de la identidad con el token del Centro de identidades de IAM para el usuario del directorio como contexto adicional, la aplicación ahora tiene todo lo necesario para realizar una solicitud firmada a `GetDataAccess` en nombre del usuario del directorio autenticado.

La propagación del token se basa en los siguientes pasos:

Crear una aplicación del Centro de identidades de IAM

En primer lugar, cree una nueva aplicación en el Centro de identidad de IAM. Esta aplicación utilizará una plantilla que permite al Centro de identidades de IAM identificar el tipo de configuración de la aplicación que puede utilizar. El comando para crear la aplicación requiere que proporciones la instancia del Centro de identidades de IAM, el nombre de recurso de Amazon (ARN), un nombre de aplicación y el ARN del proveedor de la aplicación. El proveedor de la aplicación es el proveedor de aplicaciones SAML u OAuth que la aplicación utilizará para realizar llamadas al Centro de identidades de IAM.

Para utilizar el comando de ejemplo siguiente, sustituya *`user input placeholders`* con su información.

```
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
```

Respuesta:

```
{
   "ApplicationArn": "arn:aws:sso::123456789012:application/ssoins-
ssoins-1234567890abcdef/apl-abcd1234a1b2c3d"
}
```

Crear un emisor de tokens de confianza

Ahora que dispone de la aplicación del Centro de identidades de IAM, el siguiente paso es configurar un emisor de tokens de confianza que se utilizará para intercambiar sus valores `IdToken` de su IdP por los tokens del Centro de identidades de IAM. En este paso debe proporcionar los siguientes elementos:

- La URL del emisor del proveedor de identidad
- El nombre del emisor de tokens de confianza
- La ruta del atributo de la notificación
- La ruta de los atributos del almacén de identidades
- La opción de recuperación de JSON Web Key Set (JWKS)

La ruta del atributo de la notificación es el atributo del proveedor de identidad que se utilizará para asignarse al atributo del almacén de identidades. Normalmente, la ruta del atributo de la notificación es la dirección de correo electrónico del usuario, pero puede utilizar otros atributos para realizar la asignación.

Cree un archivo de especificaciones denominado `oidc-configuration.json` con la siguiente información: Para utilizar este archivo, sustituya *`user input placeholders`* por su información.

```
{
  "OidcJwtConfiguration":
     {
      "IssuerUrl": "https://login.microsoftonline.com/a1b2c3d4-abcd-1234-b7d5-
b154440ac123/v2.0",
      "ClaimAttributePath": "preferred_username",
      "IdentityStoreAttributePath": "userName",
      "JwksRetrievalOption": "OPEN_ID_DISCOVERY"
     }
}
```

Para crear el emisor de tokens de confianza, ejecute el siguiente comando. Para utilizar este comando de ejemplo, sustituya *user input placeholders* por su propia información.

```
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
```

Respuesta

```
{
  "TrustedTokenIssuerArn": "arn:aws:sso::123456789012:trustedTokenIssuer/
ssoins-1234567890abcdef/tti-43b4a822-1234-1234-1234-a1b2c3d41234"
}
```

Conectar la aplicación del Centro de identidades de IAM con el emisor de tokens de confianza

El emisor de tokens de confianza necesita algunos ajustes de configuración más para funcionar. Defina la audiencia en la que confiará el emisor de tokens de confianza. La audiencia es el valor dentro del `IdToken` que se identifica con la clave y que se encuentra en la configuración del proveedor de identidades. Por ejemplo:

```
1234973b-abcd-1234-abcd-345c5a9c1234
```

Cree un archivo denominado `grant.json` que contenga el contenido siguiente. Para usar este archivo, cambie la audiencia para que coincida con la configuración de su proveedor de identidad y proporcione el ARN del emisor de tokens de confianza que devolvió el comando anterior.

```
{
   "JwtBearer":
     {
       "AuthorizedTokenIssuers":
         [
           {
             "TrustedTokenIssuerArn": "arn:aws:sso::123456789012:trustedTokenIssuer/
ssoins-1234567890abcdef/tti-43b4a822-1234-1234-1234-a1b2c3d41234",
               "AuthorizedAudiences":
                 [
                   "1234973b-abcd-1234-abcd-345c5a9c1234"
```

```
                ]
            }
        ]
    }
}
```

Ejecute el siguiente comando de ejemplo. Para usar este comando, sustituya *user input placeholders* por su información.

```
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 \
```

Este comando establece los ajustes de configuración para que el emisor de tokens de confianza confíe en la audiencia del archivo `grant.json` y vincule esta audiencia con la aplicación creada en el primer paso para intercambiar tokens de tipo `jwt-bearer`. La cadena `urn:ietf:params:oauth:grant-type:jwt-bearer` no es una cadena arbitraria. Es un espacio de nombres registrado en los perfiles de aserción JSON Web Token (JWT) de OAuth. Puede encontrar más información sobre este espacio de nombres en el RFC 7523.

A continuación, utilice el siguiente comando para configurar los alcances que incluirá el emisor de tokens de confianza al intercambiar valores `IdToken` desde su proveedor de identidad. En el caso de S3 Access Grants, el valor del parámetro `--scope` es `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"
```

El último paso consiste en adjuntar una política de recursos a la aplicación del Centro de identidades de IAM. Esta política permitirá al rol de IAM de su aplicación realizar solicitudes a la operación de la API `sso-oauth:CreateTokenWithIAM` y recibir los valores `IdToken` del Centro de identidades de IAM.

Cree un archivo denominado `authentication-method.json` que contenga el contenido siguiente. Reemplace *123456789012* por su ID de cuenta.

```
{
```

```
   "Iam":
       {
         "ActorPolicy":
             {
                "Version": "2012-10-17",
                    "Statement":
                    [
                        {
                           "Effect": "Allow",
                            "Principal":
                            {
                              "AWS": "arn:aws:iam::123456789012:role/webapp"
                            },
                           "Action": "sso-oauth:CreateTokenWithIAM",
                            "Resource": "*"
                        }
                    ]
                }
            }
        }
```

Para asociar la política a la aplicación del Centro de identidades de IAM, ejecute este comando:

```
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
```

Esto completa la configuración para usar S3 Access Grants con los usuarios del directorio a través de una aplicación web. Puede probar esta configuración directamente en la aplicación o puede llamar a la operación de la API `CreateTokenWithIAM` mediante el siguiente comando desde un rol de IAM permitido en la política de la aplicación del Centro de identidades de IAM:

```
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 respuesta será parecida a la siguiente:

```
{
    "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 decodifica el valor `IdToken` que está codificado con base64, puede ver los pares clave-valor en formato JSON. La clave `sts:identity_context` contiene el valor que la aplicación debe enviar en la solicitud `sts:AssumeRole` para incluir la información de identidad del usuario del directorio. A continuación, se muestra un ejemplo del `IdToken` decodificado:

```
{
    "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
}
```

Puede obtener el valor de `sts:identity_context` y pasar esta información en una llamada `sts:AssumeRole`. A continuación se muestra un ejemplo de la CLI de la sintaxis. El rol que se va a asumir es un rol temporal con permisos para invocar `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"
```

Ahora puede usar las credenciales recibidas de esta llamada para invocar la operación de la API `s3:GetDataAccess` y recibir las credenciales finales con acceso a sus recursos de S3.

Introducción a S3 Access Grants

Amazon S3 Access Grants es una característica de Amazon S3 que ofrece una solución de control de acceso escalable para sus datos de S3. S3 Access Grants es un proveedor de credenciales de S3, lo que significa que usted registra su lista de concesiones con S3 Access Grants y en qué nivel. A partir de ese momento, cuando los usuarios o clientes tengan que acceder a sus datos de S3, primero piden las credenciales a S3 Access Grants. Si existe una concesión correspondiente que autorice el acceso, S3 Access Grants proporciona credenciales de acceso temporales con privilegios mínimos. A continuación, los usuarios o clientes pueden usar las credenciales vendidas de S3 Access Grants para acceder a sus datos de S3. Teniendo esto en cuenta, si sus requisitos de datos de S3 exigen una configuración de permisos compleja o amplia, puede usar Concesiones de datos de S3 para escalar los permisos de datos de S3 para los usuarios, grupos, roles y aplicaciones.

En la mayoría de los casos de uso, puede administrar el control de acceso de sus datos de S3 mediante AWS Identity and Access Management (IAM) con políticas de bucket o políticas basadas en identidad de IAM.

No obstante, si tiene requisitos de control de acceso de S3 complejos, como los siguientes, podría beneficiarse enormemente del uso de S3 Access Grants:

- Está agotando el límite de tamaño de la política de bucket de 20 KB.
- Concede acceso a identidades humanas, por ejemplo, usuarios y grupos de Microsoft Entra ID (anteriormente Azure Active Directory), Okta o a Ping, a los datos de S3 para realizar análisis y macrodatos.
- Debe proporcionar acceso entre cuentas sin realizar actualizaciones frecuentes de las políticas de IAM.

- Los datos están sin estructurar y a nivel de objeto en lugar de estar estructurados, en formato de fila y columna.

El flujo de trabajo de S3 Access Grants es el siguiente:

Pasos	Descripción
1	Crear una instancia de S3 Access Grants Para empezar, inicie una instancia de S3 Access Grants que contenga sus concesiones de acceso individuales.
2	Registrar una ubicación En segundo lugar, registre una ubicación de datos de S3 (como la predeterminada, `s3://`) y, a continuación, especifique un rol de IAM predeterminado que S3 Access Grants asuma al brindar acceso a la ubicación de datos de S3. También puede añadir ubicaciones personalizadas a grupos o prefijos específicos y asignarlos a roles de IAM personalizados.
3	Crear concesiones Cree concesiones de permisos individuales. Especifique la ubicación S3 registrada, el alcance del acceso a los datos dentro de la ubicación, la identidad del beneficiario y su nivel de acceso (`READ`, `WRITE` o`READWRITE`) en estas concesiones de permisos.
4	Solicitar acceso a los datos de S3 Cuando los usuarios, las aplicaciones y los Servicios de AWS deseen acceder a los datos de S3, primero realizan una solicitud de acceso. S3 Access Grants determina si la solicitud debe autorizarse. Si existe una concesión correspondiente que autorice el acceso, S3 Access Grants utiliza el rol de IAM de la ubicación registrada asociado a esa concesión para devolver las credenciales temporales al solicitante.

Pasos	Descripción
5	Acceder a los datos de S3 Las aplicaciones utilizan las credenciales temporales que suministra S3 Access Grants para acceder a los datos de S3.

Crear una instancia de S3 Access Grants

Para empezar a utilizar Amazon S3 Access Grants, primero debe crear una instancia de S3 Access Grants. Solo puede crear una instancia de S3 Access Grants por Región de AWS por cuenta. La instancia de S3 Access Grants sirve como contenedor para sus recursos de S3 Access Grants, que incluyen las ubicaciones y las concesiones registradas.

Con S3 Access Grants, puede crear concesiones de permisos para sus datos de S3 para los usuarios y roles de AWS Identity and Access Management (IAM). Si ha añadido su directorio de identidades corporativas a AWS IAM Identity Center, puede asociar esta instancia del Centro de identidades de IAM de su directorio corporativo a su instancia de S3 Access Grants. Una vez que lo haya hecho, puede crear concesiones de acceso para sus usuarios y grupos corporativos. Si aún no ha añadido su directorio corporativo al Centro de identidades de IAM, puede asociar su instancia de S3 Access Grants a una instancia de este más adelante.

Puede crear una instancia de S3 Access Grants mediante la consola de Amazon S3, la AWS Command Line Interface (AWS CLI), la API de REST de Amazon S3 y los SDK de AWS.

Uso de la consola de S3

Para poder conceder acceso a sus datos de S3 con S3 Access Grants, primero debe crear una instancia de S3 Access Grants en la misma Región de AWS que sus datos de S3.

Requisitos previos

Si desea conceder acceso a sus datos de S3 mediante identidades de su directorio corporativo, añada su directorio de identidades corporativas a AWS IAM Identity Center. Si aún no está preparado para hacerlo, puede asociar su instancia de S3 Access Grants a una instancia del Centro de identidades de IAM más adelante.

Para crear una instancia de S3 Access Grants

1. Inicie sesión AWS Management Console Management Console y abra la consola de Amazon S3 en https://console.aws.amazon.com/s3/.
2. En la barra de navegación, elija el nombre de la Región de AWS que aparece. A continuación, elija la región a la que desea cambiar.
3. En el panel de navegación izquierdo, seleccione Concesiones de acceso.
4. En la página S3 Access Grants, seleccione Crear instancia de S3 Access Grants.

   a. En el paso 1 del asistente Configurar la instancia de concesiones de acceso, verifique que desea crear la instancia en la Región de AWS actual. Asegúrese de que es la misma Región de AWS donde se encuentran los datos de S3. Puede crear una instancia de S3 Access Grants por Región de AWS por cuenta.

   b. (Opcional) Si ha añadido su directorio de identidades corporativas a AWS IAM Identity Center, puede asociar esta instancia del Centro de identidades de IAM de su directorio corporativo a su instancia de S3 Access Grants.

      Para ello, seleccione Añadir una instancia del Centro de identidades de IAM en ***región***. Después, introduzca el Nombre de recurso de Amazon (ARN) de la instancia del Centro de identidades de IAM.

      Si aún no ha añadido su directorio corporativo al Centro de identidades de IAM, puede asociar su instancia de S3 Access Grants a una instancia de este más adelante.

   c. Para crear la instancia de S3 Access Grants, seleccione Siguiente. Para registrar una ubicación, consulte el Paso 2: Registrar una ubicación.

5. Si Siguiente o Crear instancia de S3 Access Grants están desactivadas:

   No se puede crear una instancia

   - Es posible que ya tenga una instancia de S3 Access Grants en la misma Región de AWS. En el panel de navegación izquierdo, seleccione Concesiones de acceso. En la página S3 Access Grants, desplácese hacia abajo hasta la sección Instancia de S3 Access Grants de su cuenta para determinar si ya existe una instancia.
   - Es posible que no tenga el permiso `s3:CreateAccessGrantsInstance` necesario para crear una instancia de S3 Access Grants. Póngase en contacto con el administrador de la cuenta. Para obtener permisos adicionales que son necesarios si se asocia una

instancia del Centro de identidades de IAM con su instancia de S3 Access Grants, consulte CreateAccessGrantsInstance.

Uso de la AWS CLI

Para instalar AWS CLI, consulte Instalación de AWS CLI en la Guía del usuario de AWS Command Line Interface.

Para utilizar el comando de ejemplo siguiente, sustituya *user input placeholders* con su información.

Example Crear una instancia de S3 Access Grants

```
aws s3control create-access-grants-instance \
--account-id 111122223333 \
--region us-east-2
```

Respuesta:

```
{
    "CreatedAt": "2023-05-31T17:54:07.893000+00:00",
    "AccessGrantsInstanceId": "default",
    "AccessGrantsInstanceArn": "arn:aws:s3:us-east-2:111122223333:access-grants/
default"
}
```

Uso de la API de REST

Puede utilizar la API de REST de Amazon S3 para crear una instancia de S3 Access Grants. Para obtener información sobre la compatibilidad con la API de REST para administrar una instancia de S3 Access Grants, consulte las siguientes secciones en la Referencia de la API de Amazon Simple Storage Service:

- AssociateAccessGrantsIdentityCenter
- CreateAccessGrantsInstance
- DeleteAccessGrantsInstance
- DissociateAccessGrantsIdentityCenter
- GetAccessGrantsInstance

- GetAccessGrantsInstanceForPrefix
- GetAccessGrantsInstanceResourcePolicy
- ListAccessGrantsInstances
- PutAccessGrantsInstanceResourcePolicy

Uso de los AWS SDK

En esta sección, se proporciona un ejemplo acerca de cómo crear una instancia de S3 Access Grants con los SDK de AWS.

Java

En este ejemplo se crea la instancia de S3 Access Grants, que sirve como contenedor para sus concesiones de acceso individuales. Puede tener una instancia de S3 Access Grants por Región de AWS en su cuenta. La respuesta incluye el ID de la instancia `default` y un nombre de recurso de Amazon (ARN) que se genera para la instancia de S3 Access Grants.

Example Crear una solicitud de instancia de S3 Access Grants

```
public void createAccessGrantsInstance() {
CreateAccessGrantsInstanceRequest createRequest =
 CreateAccessGrantsInstanceRequest.builder().accountId("111122223333").build();
CreateAccessGrantsInstanceResponse createResponse =
 s3Control.createAccessGrantsInstance(createRequest);LOGGER.info("CreateAccessGrantsInstance
 " + createResponse);
}
```

Respuesta:

```
CreateAccessGrantsInstanceResponse(
CreatedAt=2023-06-07T01:46:20.507Z,
AccessGrantsInstanceId=default,
AccessGrantsInstanceArn=arn:aws:s3:us-east-2:111122223333:access-grants/default)
```

Temas

- Ver los detalles de una instancia de S3 Access Grants
- Asociar o desasociar su instancia del Centro de identidades de IAM
- Eliminar una instancia de S3 Access Grants

Ver los detalles de una instancia de S3 Access Grants

Puede ver los detalles de su instancia de Amazon S3 Access Grants en una Región de AWS concreta. También puede enumerar sus instancias de S3 Access Grants, incluidas las instancias que se han compartido con usted a través de AWS Resource Access Manager (AWS RAM).

Puede ver los detalles de su instancia de S3 Access Grants o enumerar sus instancias de S3 Access Grants mediante la consola de Amazon S3, la AWS Command Line Interface (AWS CLI), la API de REST de Amazon S3 y los SDK de AWS.

Uso de la consola de S3

Para ver una instancia de S3 Access Grants

1. Inicie sesión AWS Management Console Management Console y abra la consola de Amazon S3 en https://console.aws.amazon.com/s3/.
2. En el panel de navegación izquierdo, seleccione Concesiones de acceso.
3. En la página S3 Access Grants, seleccione la región que contiene la instancia de S3 Access Grants con la que quiere trabajar.
4. En la página S3 Access Grants se muestran sus instancias de S3 Access Grants y cualquier instancia entre cuentas que se haya compartido con su cuenta. Para ver los detalles de una instancia, seleccione Ver detalles.

Uso de la AWS CLI

Para instalar AWS CLI, consulte Instalación de AWS CLI en la Guía del usuario de AWS Command Line Interface.

Para utilizar el comando de ejemplo siguiente, sustituya *user input placeholders* con su información.

Example — Obtener los detalles de una instancia de S3 Access Grants

```
aws s3control get-access-grants-instance \
 --account-id 111122223333 \
 --region us-east-2
```

Respuesta:

```
{
```

```
    "AccessGrantsInstanceArn": "arn:aws:s3:us-east-2: 111122223333:access-grants/
default",
    "AccessGrantsInstanceId": "default",
    "CreatedAt": "2023-05-31T17:54:07.893000+00:00"
}
```

Example — Enumerar todas las instancias de S3 Access Grants de una cuenta

Esta acción muestra las instancias de S3 Access Grants de una cuenta. Solo puede tener una instancia de S3 Access Grants por Región de AWS. Esta acción también muestra otras instancias de S3 Access Grants entre cuentas a las que tiene acceso su cuenta.

```
aws s3control list-access-grants-instances \
 --account-id 111122223333 \
 --region us-east-2
```

Respuesta:

```
{
    "AccessGrantsInstanceArn": "arn:aws:s3:us-east-2: 111122223333:access-grants/
default",
    "AccessGrantsInstanceId": "default",
    "CreatedAt": "2023-05-31T17:54:07.893000+00:00"
}
```

Uso de la API de REST

Para obtener información sobre la compatibilidad con la API de REST de Amazon S3 para administrar una instancia de S3 Access Grants, consulte las siguientes secciones en la Referencia de la API de Amazon Simple Storage Service:

- GetAccessGrantsInstance
- GetAccessGrantsInstanceForPrefix
- ListAccessGrantsInstances

Uso de los AWS SDK

En esta sección se proporcionan ejemplos de cómo obtener los detalles de una instancia de S3 Access Grants con los SDK de AWS.

Para utilizar los siguientes ejemplos, reemplace los *user input placeholders* con su propia información.

Java

Example — Obtener una instancia de S3 Access Grants

```
public void getAccessGrantsInstance() {
GetAccessGrantsInstanceRequest getRequest = GetAccessGrantsInstanceRequest.builder()
.accountId("111122223333")
.build();
GetAccessGrantsInstanceResponse getResponse =
 s3Control.getAccessGrantsInstance(getRequest);
LOGGER.info("GetAccessGrantsInstanceResponse: " + getResponse);
}
```

Respuesta:

```
GetAccessGrantsInstanceResponse(
AccessGrantsInstanceArn=arn:aws:s3:us-east-2: 111122223333:access-grants/default,
CreatedAt=2023-06-07T01:46:20.507Z)
```

Example — Enumerar todas las instancias de S3 Access Grants de una cuenta

Esta acción muestra las instancias de S3 Access Grants de una cuenta. Solo puede tener una instancia de S3 Access Grants por región. Esta acción también puede mostrar otras instancias de S3 Access Grants entre cuentas a las que tiene acceso su cuenta.

```
public void listAccessGrantsInstances() {
ListAccessGrantsInstancesRequest listRequest =
 ListAccessGrantsInstancesRequest.builder()
.accountId("111122223333")
.build();
ListAccessGrantsInstancesResponse listResponse =
 s3Control.listAccessGrantsInstances(listRequest);
LOGGER.info("ListAccessGrantsInstancesResponse: " + listResponse);
}
```

Respuesta:

```
ListAccessGrantsInstancesResponse(
```

```
AccessGrantsInstancesList=[
ListAccessGrantsInstanceEntry(
AccessGrantsInstanceId=default,
AccessGrantsInstanceArn=arn:aws:s3:us-east-2:111122223333:access-grants/default,
CreatedAt=2023-06-07T04:28:11.728Z
)
]
)
```

Asociar o desasociar su instancia del Centro de identidades de IAM

En Amazon S3 Access Grants, puede asociar la instancia del AWS IAM Identity Center de su directorio de identidad corporativa a una instancia de S3 Access Grants. Después de hacerlo, puede crear concesiones de acceso para los usuarios y grupos de su directorio corporativo, además de los usuarios y roles de AWS Identity and Access Management (IAM).

Si ya no desea crear concesiones de acceso para los usuarios y grupos de su directorio corporativo, puede desasociar su instancia del Centro de identidades de IAM de su instancia de S3 Access Grants.

Puede asociar o desasociar una instancia del Centro de identidades de IAM mediante la consola de Amazon S3, la AWS Command Line Interface (AWS CLI), la API de REST de Amazon S3 y los SDK de AWS.

Uso de la consola de S3

Antes de asociar su instancia del Centro de identidades de IAM a su instancia de S3 Access Grants, debe añadir su directorio de identidades corporativas al Centro de identidades de IAM. Para obtener más información, consulte the section called “S3 Access Grants e identidades de directorios corporativos”.

Para asociar una instancia del Centro de identidades de IAM a una instancia de S3 Access Grants

1. Inicie sesión AWS Management Console Management Console y abra la consola de Amazon S3 en https://console.aws.amazon.com/s3/.
2. En el panel de navegación izquierdo, seleccione Concesiones de acceso.
3. En la página S3 Access Grants, seleccione la región que contiene la instancia de S3 Access Grants con la que quiere trabajar.
4. Seleccione Ver detalles de la instancia.

5. En la página de detalles, en la sección Centro de identidades de IAM, elija entre Añadir una instancia del Centro de identidades de IAM o Anular el registro de una instancia del centro de identidades de IAM ya asociada.

Uso de la AWS CLI

Para instalar AWS CLI, consulte Instalación de AWS CLI en la Guía del usuario de AWS Command Line Interface.

Para utilizar el comando de ejemplo siguiente, sustituya *user input placeholders* con su información.

Example – Asociar una instancia del Centro de identidades de IAM a una instancia de S3 Access Grants

```
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 – Desasociar una instancia del Centro de identidades de IAM a una instancia de S3 Access Grants

```
aws s3control dissociate-access-grants-identity-center \
 --account-id 111122223333 \
 --profile access-grants-profile \
 --region eu-central-1

// No response body
```

Uso de la API de REST

Para obtener información sobre la compatibilidad con la API de REST de Amazon S3 para administrar la asociación entre una instancia del Centro de identidades de IAM y una instancia de S3 Access Grants, consulte las siguientes secciones en la Referencia de la API de Amazon Simple Storage Service:

- AssociateAccessGrantsIdentityCenter
- DissociateAccessGrantsIdentityCenter

Eliminar una instancia de S3 Access Grants

Puede eliminar una instancia de S3 Access Grants de una Región de AWS de su cuenta. Sin embargo, antes de eliminar una instancia de S3 Access Grants, primero debe hacer lo siguiente:

- Elimine todos los recursos de la instancia de S3 Access Grants, incluidas todas las concesiones y ubicaciones. Para obtener más información, consulte Eliminación de una concesión y Eliminación de una ubicación.
- Si ha asociado una instancia del AWS IAM Identity Center a su instancia de S3 Access Grants, debe desasociar la instancia del Centro de identidades de IAM. Para obtener más información, consulte Asociar o desasociar su instancia del Centro de identidades de IAM.

> **Important**
>
> Si elimina una instancia de S3 Access Grants, la eliminación es permanente y no se puede deshacer. Todos los beneficiarios a los que se haya concedido acceso mediante las concesiones de esta instancia de S3 Access Grants perderán el acceso a sus datos de S3.

Puede eliminar una instancia de S3 Access Grants mediante la consola de Amazon S3, la AWS Command Line Interface (AWS CLI), la API de REST de Amazon S3 y los SDK de AWS.

Uso de la consola de S3

Para eliminar una instancia de S3 Access Grants

1. Inicie sesión AWS Management Console Management Console y abra la consola de Amazon S3 en https://console.aws.amazon.com/s3/.
2. En el panel de navegación izquierdo, seleccione Concesiones de acceso.
3. En la página S3 Access Grants, seleccione la región que contiene la instancia de S3 Access Grants con la que quiere trabajar.
4. Seleccione Ver detalles de la instancia.
5. En la página de detalles de la instancia, seleccione Eliminar instancia en la esquina superior derecha.

6. En el cuadro de diálogo que aparece, seleccione Confirmar eliminación. Esta acción no se puede deshacer.

Uso de la AWS CLI

Para instalar AWS CLI, consulte Instalación de AWS CLI en la Guía del usuario de AWS Command Line Interface.

Para utilizar el comando de ejemplo siguiente, sustituya *user input placeholders* con su información.

> **Note**
>
> Para poder eliminar una instancia de S3 Access Grants, primero debe eliminar todas las concesiones y ubicaciones creadas en la instancia de S3 Access Grants. Si ha asociado una instancia central del Centro de identidades de IAM a su instancia de S3 Access Grants, primero debe desasociarla.

Example — Eliminar una instancia de S3 Access Grants

```
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
```

Uso de la API de REST

Para obtener información sobre la compatibilidad con la API de REST de Amazon S3 para eliminar una instancia de S3 Access Grants, consulte DeleteAccessGrantsInstance en la Referencia de la API de Amazon Simple Storage Service.

Uso de los AWS SDK

En esta sección se proporcionan ejemplos de cómo eliminar una instancia de S3 Access Grants con los SDK de AWS.

Para utilizar el ejemplo siguiente, sustituya *user input placeholders* con su propia información.

Java

> **Note**
>
> Para poder eliminar una instancia de S3 Access Grants, primero debe eliminar todas las concesiones y ubicaciones creadas en la instancia de S3 Access Grants. Si ha asociado una instancia central del Centro de identidades de IAM a su instancia de S3 Access Grants, primero debe desasociarla.

Example — Eliminar una instancia de S3 Access Grants

```
public void deleteAccessGrantsInstance() {
DeleteAccessGrantsInstanceRequest deleteRequest =
 DeleteAccessGrantsInstanceRequest.builder()
.accountId("111122223333")
.build();
DeleteAccessGrantsInstanceResponse deleteResponse =
 s3Control.deleteAccessGrantsInstance(deleteRequest);
LOGGER.info("DeleteAccessGrantsInstanceResponse: " + deleteResponse);
}
```

Registrar una ubicación

Tras crear una instancia de Concesiones de acceso a Amazon S3 en una Región de AWS de su cuenta, puede registrar una ubicación de S3 en esa instancia. Una ubicación de Concesiones de acceso a Amazon S3 asigna la ubicación predeterminada de S3 (`s3://`), un bucket o un prefijo a un rol de AWS Identity and Access Management (IAM). Concesiones de acceso a Amazon S3 asume este rol de IAM para vender credenciales temporales al beneficiario que accede a esa ubicación en particular. En primer lugar, debe registrar al menos una ubicación en su instancia de Concesiones de acceso a Amazon S3 para poder crear una concesión de acceso.

Caso de uso recomendado

Le recomendamos que registre la ubicación predeterminada (`s3://`) y la asigne a un rol de IAM. La ubicación en la ruta de S3 predeterminada (`s3://`) cubre el acceso a todos los buckets de S3 en esa Región de AWS de su cuenta. Al crear una concesión de acceso, puede limitar el ámbito de la concesión a un bucket, un prefijo o un objeto dentro de la ubicación predeterminada.

Casos de uso de administración de acceso complejos

Los casos de uso de administración de acceso más complejos pueden requerir que registre una ubicación diferente a la predeterminada. Estos son algunos ejemplos de estos casos de uso:

- Supongamos que *amzn-s3-demo-bucket* es una ubicación registrada en su instancia de Concesiones de acceso a Amazon S3 con un rol de IAM asignado, pero a este rol de IAM se le niega el acceso a un prefijo concreto del bucket. En este caso, puede registrar el prefijo al que el rol de IAM no tiene acceso como una ubicación independiente y asignar esa ubicación a un rol de IAM diferente con el acceso necesario.
- Supongamos que desea crear concesiones que restrinjan el acceso solo a los usuarios dentro de un punto de conexión a la nube privada virtual (VPC). En este caso, puede registrar una ubicación para un bucket en el que el rol de IAM restrinja el acceso al punto de conexión de VPC. Más adelante, cuando un beneficiario solicite las credenciales a Concesiones de acceso a Amazon S3, este asumirá el rol de IAM de la ubicación para vender las credenciales temporales. Esta credencial denegará el acceso al bucket específico, a menos que el iniciador se encuentre dentro del punto de conexión de VPC. Esta denegación de permiso se aplica además del permiso normal READ, WRITE o READWRITE que se especifica en la concesión.

Al registrar una ubicación, también debe especificar el rol de IAM que Concesiones de acceso a Amazon S3 asume para vender credenciales temporales y limitar el ámbito de los permisos para una concesión específica.

Si su caso de uso requiere que registre varias ubicaciones en su instancia de Concesiones de acceso a Amazon S3, puede registrar cualquiera de las siguientes opciones:

S3 URI	Rol de IAM	Descripción
`s3://`	*Default-IAM-role*	La ubicación predeterminada, `s3://`, incluye todos los buckets de Región de AWS.
`s3://`*amzn-s3-demo-bucket1* `/`	*IAM-role-For-bucket*	En esta ubicación se incluyen todos los objetos del bucket especificado.
`s3://`*amzn-s3-demo-bucket1* `/`*prefix-name*	*IAM-role-For-prefix*	En esta ubicación se incluyen todos los objetos del bucket con un nombre de clave de objeto que comience con este prefijo.

Antes de poder registrar un bucket o prefijo específicos, asegúrese de que hace lo siguiente:

- Cree uno o varios buckets que contengan los datos a los que desea otorgar acceso. Estos bucket deben encontrarse en la misma Región de AWS que su instancia de S3 Access Grants. Para obtener más información, consulte Creación de un bucket.

  La adición de un prefijo es un paso opcional. Los prefijos son cadenas al principio del nombre de la clave del objeto. Puede utilizarlos para organizar los objetos en su bucket, así como para administrar el acceso. Para añadir un prefijo a un bucket, consulte Creación de nombres de clave de objeto.
- Cree un rol de IAM que tenga permisos para acceder a sus datos de S3 en la Región de AWS. Para obtener más información, consulte Creación de roles de IAM en la AWS IAM Identity CenterGuía del usuario de .
- En la política de confianza del rol de IAM, proporcione al servicio de Concesiones de acceso a Amazon S3 acceso de entidad principal (`access-grants.s3.amazonaws.com`) al rol de IAM que ha creado. Para ello, puede crear un archivo JSON que contenga las siguientes instrucciones. Para agregar la política de confianza a su cuenta, consulte Creación de un rol mediante políticas de confianza personalizadas.

  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"},
   "Condition":{
  "StringEquals":{
   "aws:SourceAccount":"accountId",
   "aws:SourceArn":"arn:aws:s3:region:accountId:access-grants/default"
  }
  //Optionally, for an IAM Identity Center use case, add:
  "ForAnyValue:StringEquals": {
   "aws:RequestContextProvider":"arn:aws:iam::aws:contextProvider/IdentityCenter"
  }
 }
    }
  ]
```

```
}
```

- Cree una política de IAM para asociar permisos de Amazon S3 al rol de IAM que ha creado. Consulte el siguiente archivo `iam-policy.json` de ejemplo y sustituya *`user input placeholders`* por su propia información.

> **Note**
>
> - Si utiliza el cifrado del servidor con claves de AWS Key Management Service (AWS KMS) para cifrar sus datos, en el siguiente ejemplo se incluyen los permisos de AWS KMS necesarios para el rol de IAM de la política. Si no utiliza esta característica, puede eliminar estos permisos de su política de IAM.
> - Puede restringir el acceso del rol de IAM a los datos de S3 solo si S3 Access Grants ofrece las credenciales. En este ejemplo se muestra cómo añadir una instrucción `Condition` para una instancia de S3 Access Grants. Para usar esta `Condition`, sustituya el ARN de instancia de Concesiones de acceso a Amazon S3 en la instrucción `Condition` por el ARN de instancia de Concesiones de acceso a Amazon S3, que tiene el siguiente formato: `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:Región de
 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:Región de
 AWS:accountId:access-grants/default"]
            }
```

```
            }
        },
    //Optionally add the following section if you use SSE-KMS encryption
        {
            "Sid": "KMSPermissions",
            "Effect":"Allow",
            "Action":[
                "kms:Decrypt",
                "kms:GenerateDataKey"
            ],
            "Resource":[
                "*"
            ]
        }
    ]
}
```

Puede registrar una ubicación en su instancia de S3 Access Grants mediante la consola de Amazon S3, la AWS Command Line Interface (AWS CLI), la API de REST de Amazon S3 o los SDK de AWS.

> **Note**
>
> Tras registrar la primera ubicación en su instancia de Concesiones de acceso a Amazon S3, la instancia seguirá sin tener ninguna concesión de acceso individual. Para crear una concesión de acceso, consulte Crear concesiones.

Uso de la consola de S3

Debe tener al menos una ubicación registrada antes de poder otorgar acceso a sus datos de S3 con S3 Access Grants.

Para registrar una ubicación en su instancia de S3 Access Grants

1. Inicie sesión AWS Management Console Management Console y abra la consola de Amazon S3 en https://console.aws.amazon.com/s3/.
2. En el panel de navegación izquierdo, seleccione Concesiones de acceso.
3. En la página S3 Access Grants, seleccione la región que contiene la instancia de S3 Access Grants con la que quiere trabajar.

Si es la primera vez que utiliza la instancia de S3 Access Grants, asegúrese de haber completado el Paso 1: Crear una instancia de S3 Access Grants) y de haber navegado al Paso 2 del asistente Configurar la instancia de Concesiones de acceso. Si ya tiene una instancia de S3 Access Grants, seleccione Ver detalles y, en la pestaña Ubicaciones, elija Registrar ubicación.

a. En Ámbito de ubicación, elija Examinar S3 o introduzca la ruta URI de S3 a la ubicación que desee registrar. Para ver los formatos de URI de S3, consulte la tabla de formatos de ubicación. Tras introducir un URI, puede seleccionar Ver para navegar hasta la ubicación.
b. En Rol de IAM, elija una de las siguientes opciones:

   - Elija entre los roles de IAM existentes

     Elija un rol de IAM de la lista desplegable. Después de elegir un rol, elija Ver para asegurarse de que este rol tiene los permisos necesarios para administrar la ubicación que está registrando. En concreto, asegúrese de que este rol conceda a S3 Access Grants los permisos `sts:AssumeRole` `ysts:SetSourceIdentity`.
   - Introduzca el ARN del rol de IAM

     Vaya a la consola de IAM. Copie el Nombre de recurso de Amazon (ARN) del rol de IAM y péguelo en este cuadro.
c. Para terminar, seleccione Siguiente o Registrar ubicación.

4. Solución de problemas:

   No se puede registrar la ubicación

   - Es posible que la ubicación ya esté registrada.

     Puede que no tenga el permiso `s3:CreateAccessGrantsLocation` para registrar ubicaciones. Póngase en contacto con el administrador de la cuenta.

Uso de la AWS CLI

Para instalar AWS CLI, consulte Instalación de AWS CLI en la Guía del usuario de AWS Command Line Interface.

Puede registrar la ubicación predeterminada, `s3://`, o una ubicación personalizada en su instancia de S3 Access Grants. Asegúrese de crear primero un rol de IAM con acceso de entidad principal a la ubicación y, después, de conceder el permiso a S3 Access Grants para asumir este rol.

Para utilizar los comandos de ejemplo siguientes, sustituya *user input placeholders* con su información.

Example Crear una política de recursos

Cree una política que permita que S3 Access Grants pueda asumir el rol de IAM. Para ello, puede crear un archivo JSON que contenga las siguientes instrucciones. Para añadir la política de recursos a su cuenta, consulte Crear y asociar su primera política administrada por el cliente.

TestRolePolicy.json

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Sid": "Stmt1234567891011",
      "Action": ["sts:AssumeRole", "sts:SetSourceIdentity"],
      "Effect": "Allow",
      "Principal": {"Service":"access-grants.s3.amazonaws.com"}
    }
  ]
}
```

Example Crear el rol

Ejecute el siguiente comando de IAM para crear el rol.

```
aws iam create-role --role-name accessGrantsTestRole \
 --region us-east-2 \
 --assume-role-policy-document file://TestRolePolicy.json
```

Al ejecutar el comando `create-role` se devuelve la política:

```
{
    "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

Crear una política de IAM para asociar permisos de Amazon S3 al rol de IAM. Consulte el siguiente archivo `iam-policy.json` de ejemplo y sustituya *`user input placeholders`* por su propia información.

> **Note**
>
> Si utiliza el cifrado del servidor con claves de AWS Key Management Service (AWS KMS) para cifrar sus datos, en el siguiente ejemplo se añaden los permisos de AWS KMS necesarios para el rol de IAM de la política. Si no utiliza esta característica, puede eliminar estos permisos de su política de IAM.
>
> Para garantizar que el rol de IAM solo se pueda utilizar para acceder a los datos de S3 en caso de que S3 Access Grants proporcione las credenciales, en este ejemplo se muestra cómo añadir una instrucción `Condition` que especifique la instancia de S3 Access Grants (`s3:AccessGrantsInstance:` *`InstanceArn`*) en su política de IAM. Al utilizar la siguiente política de ejemplo, sustituya *`user input placeholders`* con su propia información.

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:Región de
 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:Región de
 AWS:accountId:access-grants/default"]
            }
         }
      },
      {
         "Sid": "KMSPermissions",
         "Effect":"Allow",
         "Action":[
            "kms:Decrypt",
            "kms:GenerateDataKey"
         ],
         "Resource":[
            "*"
         ]
      }
   ]
}
```

Example

Ejecute el siguiente comando:

```
aws iam put-role-policy \
--role-name accessGrantsTestRole \
--policy-name accessGrantsTestRole \
--policy-document file://iam-policy.json
```

Example Registrar la ubicación predeterminada

```
aws s3control create-access-grants-location \
 --account-id 111122223333 \
 --location-scope s3:// \
 --iam-role-arn arn:aws:iam::111122223333:role/accessGrantsTestRole
```

Respuesta:

```
{"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 Registrar una ubicación personalizada

```
aws s3control create-access-grants-location \
 --account-id 111122223333 \
 --location-scope s3://DOC-BUCKET-EXAMPLE/ \
 --iam-role-arn arn:aws:iam::123456789012:role/accessGrantsTestRole
```

Respuesta:

```
{"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"
}
```

Uso de la API de REST

Para obtener información sobre la compatibilidad con la API de REST de Amazon S3 para administrar una instancia de S3 Access Grants, consulte las siguientes secciones en la Referencia de la API de Amazon Simple Storage Service:

- CreateAccessGrantsLocation

- DeleteAccessGrantsLocation
- GetAccessGrantsLocation
- ListAccessGrantsLocations
- UpdateAccessGrantsLocation

Uso de los AWS SDK

En esta sección se proporcionan ejemplos de cómo registrar ubicaciones con los SDK de AWS.

Para utilizar los siguientes ejemplos, reemplace los *user input placeholders* con su propia información.

Java

Puede registrar la ubicación predeterminada, `s3://`, o una ubicación personalizada en su instancia de S3 Access Grants. Asegúrese de crear primero un rol de IAM con acceso de entidad principal a la ubicación y, después, de conceder el permiso a S3 Access Grants para asumir este rol.

Para utilizar los comandos de ejemplo siguientes, sustituya *user input placeholders* con su información.

Example Registrar una ubicación predeterminada

Solicitud:

```
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);
}
```

Response: (Respuesta:)

```
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 Registrar una ubicación personalizada

Solicitud:

```
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);
}
```

Response: (Respuesta:)

```
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
)
```

Temas

- Ver los detalles de una ubicación registrada
- Actualizar una ubicación registrada
- Eliminar una ubicación registrada

Ver los detalles de una ubicación registrada

Puede obtener los detalles de una ubicación en su instancia de S3 Access Grants mediante la consola de Amazon S3, la AWS Command Line Interface (AWS CLI), la API de REST de Amazon S3 y los SDK de AWS.

Uso de la consola de S3

Para ver las ubicaciones registradas en su instancia de S3 Access Grants

1. Inicie sesión AWS Management Console Management Console y abra la consola de Amazon S3 en https://console.aws.amazon.com/s3/.
2. En el panel de navegación izquierdo, seleccione Concesiones de acceso.
3. En la página S3 Access Grants, seleccione la región que contiene la instancia de S3 Access Grants con la que quiere trabajar.
4. Seleccione Ver detalles de la instancia.
5. En la página de detalles de la instancia, elija Ubicaciones.
6. Busque la ubicación registrada que desea ver. Utilice el cuadro de búsqueda para filtrar la lista de ubicaciones registradas.

Uso de la AWS CLI

Para instalar AWS CLI, consulte Instalación de AWS CLI en la Guía del usuario de AWS Command Line Interface.

Para utilizar el comando de ejemplo siguiente, sustituya *user input placeholders* con su información.

Example – Obtener los detalles de una ubicación registrada

```
aws s3control get-access-grants-location \
--account-id 111122223333 \
--access-grants-location-id default
```

Respuesta:

```
{
    "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 – Enumerar todas las ubicaciones que están registradas en una instancia de S3 Access Grants

Para restringir los resultados a un prefijo o bucket de S3, puede utilizar el parámetro `--location-scope s3://bucket-and-or-prefix` si lo desea.

```
aws s3control list-access-grants-locations \
--account-id 111122223333 \
--region us-east-2
```

Respuesta:

```
{"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://amzn-s3-demo-bucket/prefixA*",
    "IAMRoleArn": "arn:aws:iam::111122223333:role/accessGrantsTestRole"
     }
   ]
  }
```

Uso de la API de REST

Para obtener información sobre la compatibilidad de la API de REST de Amazon S3 a fin de obtener los detalles de una ubicación registrada o enumerar todas las ubicaciones que están registradas en

una instancia de S3 Access Grants, consulte las siguientes secciones de la Referencia de la API de Amazon Simple Storage Service:

- GetAccessGrantsLocation
- ListAccessGrantsLocations

Uso de los AWS SDK

En esta sección se proporcionan ejemplos de cómo obtener los detalles de una ubicación registrada o enumerar todas las ubicaciones registradas en una instancia de S3 Access Grants mediante los SDK de AWS.

Para utilizar los siguientes ejemplos, reemplace los *`user input placeholders`* con su propia información.

Java

Example – Obtener los detalles de una ubicación registrada

```
public void getAccessGrantsLocation() {
GetAccessGrantsLocationRequest getAccessGrantsLocationRequest =
 GetAccessGrantsLocationRequest.builder()
.accountId("111122223333")
.accessGrantsLocationId("default")
.build();
GetAccessGrantsLocationResponse getAccessGrantsLocationResponse =
 s3Control.getAccessGrantsLocation(getAccessGrantsLocationRequest);
LOGGER.info("GetAccessGrantsLocationResponse: " + getAccessGrantsLocationResponse);
}
```

Respuesta:

```
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 – Enumerar todas las ubicaciones registradas en una instancia de S3 Access Grants

Para restringir los resultados a un prefijo o bucket de S3, si lo desea, puede pasar un URI de S3, por ejemplo `s3://bucket-and-or-prefix`, en el parámetro `LocationScope`.

```
public void listAccessGrantsLocations() {

ListAccessGrantsLocationsRequest listRequest =
 ListAccessGrantsLocationsRequest.builder()
.accountId("111122223333")
.build();

ListAccessGrantsLocationsResponse listResponse =
 s3Control.listAccessGrantsLocations(listRequest);
LOGGER.info("ListAccessGrantsLocationsResponse: " + listResponse);
}
```

Respuesta:

```
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://amzn-s3-demo-bucket/prefixA*,
IAMRoleArn=arn:aws:iam::111122223333:role/accessGrantsTestRole
)
]
)
```

Actualizar una ubicación registrada

Puede actualizar el rol de AWS Identity and Access Management (IAM) de una ubicación que está registrada en su instancia de Amazon S3 Access Grants. Para cada nuevo rol de IAM que utilice para registrar una ubicación en S3 Access Grants, asegúrese de conceder acceso a la entidad de servicio de S3 Access Grants (`access-grants.s3.amazonaws.com`) a este rol. Para ello, añada una entrada para el nuevo rol de IAM en el mismo archivo JSON de política de confianza que utilizó al registrar la ubicación por primera vez.

Puede actualizar una ubicación en su instancia de S3 Access Grants mediante la consola de Amazon S3, la AWS Command Line Interface (AWS CLI), la API de REST de Amazon S3 y los SDK de AWS.

Uso de la consola de S3

Para actualizar el rol de IAM de una ubicación registrada en su instancia de S3 Access Grants

1. Inicie sesión AWS Management Console Management Console y abra la consola de Amazon S3 en https://console.aws.amazon.com/s3/.
2. En el panel de navegación izquierdo, seleccione Concesiones de acceso.
3. En la página S3 Access Grants, seleccione la región que contiene la instancia de S3 Access Grants con la que quiere trabajar.
4. Seleccione Ver detalles de la instancia.
5. En la página de detalles de la instancia, elija Ubicaciones.
6. Encuentre la ubicación que desea actualizar. Para filtrar la lista de ubicaciones, utilice el cuadro de búsqueda.
7. Elija el botón de opciones situado al lado de la ubicación registrada que desea actualizar.
8. Actualice el rol de IAM y, a continuación, seleccione Guardar cambios.

Uso de la AWS CLI

Para instalar AWS CLI, consulte Instalación de AWS CLI en la Guía del usuario de AWS Command Line Interface.

Para utilizar el comando de ejemplo siguiente, sustituya *`user input placeholders`* con su información.

Example – Actualizar el rol de IAM de una ubicación registrada

```
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
```

Respuesta:

```
{
    "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://amzn-s3-demo-bucket/prefixB*",
    "IAMRoleArn": "arn:aws:iam::777788889999:role/accessGrantsTestRole"
}
```

Uso de la API de REST

Para obtener información sobre la compatibilidad con la API de REST de Amazon S3 para actualizar una ubicación en una instancia de S3 Access Grants, consulte UpdateAccessGrantsLocation en la Referencia de la API de Amazon Simple Storage Service.

Uso de los AWS SDK

En esta sección se proporcionan ejemplos de cómo actualizar el rol de IAM de una ubicación registrada con los SDK de AWS.

Para utilizar el ejemplo siguiente, sustituya *user input placeholders* con su propia información.

Java

Example – Actualizar el rol de IAM de una ubicación registrada

```
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);
}
```

Respuesta:

```
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://amzn-s3-demo-bucket/prefixB*,
IAMRoleArn=arn:aws:iam::777788889999:role/accessGrantsTestRole
)
```

Eliminar una ubicación registrada

Puede eliminar el registro de una ubicación de una instancia de Amazon S3 Access Grants. Al eliminar la ubicación, se anula su registro de la instancia de S3 Access Grants.

Para poder eliminar el registro de una ubicación de una instancia de S3 Access Grants, debe eliminar todas las concesiones asociadas a esta ubicación. Para obtener información sobre cómo eliminar concesiones, consulte Eliminar una concesión.

Puede eliminar una ubicación en su instancia de S3 Access Grants mediante la consola de Amazon S3, la AWS Command Line Interface (AWS CLI), la API de REST de Amazon S3 y los SDK de AWS.

Uso de la consola de S3

Para eliminar el registro de una ubicación de su instancia de Amazon S3 Access Grants

1. Inicie sesión AWS Management Console Management Console y abra la consola de Amazon S3 en https://console.aws.amazon.com/s3/.
2. En el panel de navegación izquierdo, seleccione Concesiones de acceso.
3. En la página S3 Access Grants, seleccione la región que contiene la instancia de S3 Access Grants con la que quiere trabajar.
4. Seleccione Ver detalles de la instancia.
5. En la página de detalles de la instancia, elija Ubicaciones.

6. Encuentre la ubicación que desea actualizar. Para filtrar la lista de ubicaciones, utilice el cuadro de búsqueda.
7. Elija el botón de opciones situado al lado de la ubicación registrada que desea eliminar.
8. Elija Anular registro.
9. Aparece un cuadro de diálogo que le advierte de que esta acción no se puede deshacer. Para eliminar la ubicación, elija Anular registro.

Uso de la AWS CLI

Para instalar AWS CLI, consulte Instalación de AWS CLI en la Guía del usuario de AWS Command Line Interface.

Para utilizar el comando de ejemplo siguiente, sustituya *user input placeholders* con su información.

Example – Eliminar el registro de una ubicación

```
aws s3control delete-access-grants-location \
--account-id 111122223333 \
--access-grants-location-id  a1b2c3d4-5678-90ab-cdef-EXAMPLE11111
 // No response body
```

Uso de la API de REST

Para obtener información sobre la compatibilidad con la API de REST de Amazon S3 para eliminar una ubicación de una instancia de S3 Access Grants, consulte DeleteAccessGrantsLocation en la Referencia de la API de Amazon Simple Storage Service.

Uso de los AWS SDK

En esta sección se proporciona un ejemplo acerca de cómo eliminar una ubicación con los SDK de AWS.

Para utilizar el ejemplo siguiente, sustituya *user input placeholders* con su propia información.

Java

Example – Eliminar el registro de una ubicación

```
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);
}
```

Respuesta:

```
DeleteAccessGrantsLocationResponse()
```

Crear concesiones

Una concesión de acceso individual en una instancia de Concesiones de acceso a Amazon S3 permite que una identidad específica (una entidad principal de AWS Identity and Access Management [IAM] o un usuario o grupo de un directorio corporativo) acceda a una ubicación que esté registrada en su instancia de Concesiones de acceso a Amazon S3. Una ubicación asigna buckets o prefijos a un rol de IAM. Concesiones de acceso a Amazon S3 asume este rol de IAM para vender credenciales temporales a los beneficiarios.

Después de registrar al menos una ubicación en su instancia de Concesiones de acceso a Amazon S3, puede crear una concesión de acceso.

El beneficiario puede ser un usuario o rol de IAM o un usuario o grupo del directorio. Un usuario de directorio es un usuario de su directorio corporativo u origen de identidad externa que ha asociado a su instancia de Concesiones de acceso a Amazon S3. Para obtener más información, consulte S3 Access Grants e identidades de directorios corporativos. Para crear una concesión para un usuario o grupo de directorio específico desde IAM Identity Center, busque el GUID que IAM Identity Center utiliza para identificar a ese usuario en IAM Identity Center (por ejemplo, `a1b2c3d4-5678-90ab-cdef-EXAMPLE11111`). Para obtener más información sobre cómo utilizar IAM Identity Center para ver la información de los usuarios, consulte View user and group assignments en la Guía del usuario de AWS IAM Identity Center.

Puede conceder acceso a un bucket, a un prefijo o a un objeto. Un prefijo en Amazon S3 es una cadena de caracteres al principio del nombre de una clave de objeto que se utiliza para organizar

los objetos dentro de un bucket. Puede ser cualquier cadena de caracteres permitidos, por ejemplo, nombres de claves de objetos en un bucket que comiencen con el prefijo `engineering/`.

Subprefijo

Al conceder acceso a una ubicación registrada, puede usar el campo `Subprefix` para limitar el ámbito del acceso a un subconjunto del ámbito de la ubicación. Si la ubicación registrada que elija para la concesión es la ruta de S3 predeterminada (`s3://`), debe limitar el ámbito de la concesión. No puede crear una concesión de acceso para la ubicación predeterminada (`s3://`), porque daría al beneficiario acceso a todos los buckets de una Región de AWS. En su lugar, debe limitar el ámbito de la concesión a uno de los siguientes:

- Un bucket: `s3://`*`bucket`*`/*`
- Un prefijo dentro de un bucket: `s3://`*`bucket`*`/`*`prefix`*`*`
- Un prefijo dentro de un prefijo: `s3://`*`bucket`*`/`*`prefixA`*`/`*`prefixB`*`*`
- Un objeto: `s3://`*`bucket`*`/`*`object-key-name`*

Si crea una concesión de acceso en la que la ubicación registrada es un bucket, puede pasar una de las siguientes opciones en el campo `Subprefix` para limitar el ámbito de la concesión:

- Un prefijo dentro del bucket: *`prefix`*`*`
- Un prefijo dentro de un prefijo: *`prefixA/prefixB`*`*`
- Un objeto: `/`*`object-key-name`*

Después de crear la concesión, el ámbito de la concesión que se muestra en la consola de Amazon S3 o el `GrantScope` que se devuelve en la respuesta de la API o la AWS Command Line Interface (AWS CLI) es el resultado de concatenar la ruta de ubicación con el `Subprefix`. Asegúrese de que esta ruta concatenada se asigne correctamente al bucket, prefijo u objeto de S3 al que desea conceder acceso.

> **Note**
>
> - Si necesita crear una concesión de acceso que conceda acceso a un solo objeto, debe especificar que el tipo de concesión es para un objeto. Para hacerlo en una llamada a la API o un comando de la CLI, pase el parámetro `s3PrefixType` con el valor`Object`. En la

consola de Amazon S3, al crear la concesión, tras seleccionar una ubicación, en Ámbito de la concesión, active la casilla El ámbito de la concesión es un objeto.

- No puede crear una concesión para un bucket si el bucket aún no existe. Sin embargo, puede crear una concesión para un prefijo que aún no exista.
- Para obtener información sobre el número máximo de concesiones que puede crear en su instancia de Concesiones de acceso a Amazon S3, consulte Limitaciones de S3 Access Grants.

Puede crear una concesión de acceso a través de la consola de Amazon S3, AWS CLI, la API de REST de Amazon S3 y los SDK de AWS.

Uso de la consola de S3

Para crear una concesión de acceso

1. Inicie sesión AWS Management Console Management Console y abra la consola de Amazon S3 en https://console.aws.amazon.com/s3/.
2. En el panel de navegación izquierdo, seleccione Concesiones de acceso.
3. En la página S3 Access Grants, seleccione la región que contiene la instancia de S3 Access Grants con la que quiere trabajar.

   Si es la primera vez que utiliza la instancia de S3 Access Grants, asegúrese de haber completado el Paso 2: Registrar una ubicación) y de haber navegado al Paso 3 del asistente Configurar la instancia de Concesiones de acceso. Si ya tiene una instancia de S3 Access Grants, seleccione Ver detalles y, en la pestaña Concesiones, elija Crear concesión.

   a. En la sección Alcance de la concesión, seleccione o introduzca una ubicación registrada.

      Si se ha seleccionado la ubicación predeterminada `s3://`, utilice el cuadro Subprefijo para reducir el alcance de la concesión de acceso. Para obtener más información, consulte Subprefijo. Si concede acceso solo a un objeto, seleccione Otorgar el alcance a un objeto.

   b. En Permisos y acceso, seleccione el nivel de Permiso, ya sea de Lectura, Escritura o ambos.

      A continuación, elija el Tipo de beneficio. Si ha añadido su directorio corporativo al Centro de identidades de IAM y ha asociado esta instancia del Centro de identidades de IAM a su instancia de S3 Access Grants, puede elegir Identidad del directorio en el Centro de

identidades de IAM. Si elige esta opción, obtenga el ID del usuario o grupo del Centro de identidades de IAM e introdúzcalo en esta sección.

Si el Tipo de beneficiario es un usuario o un rol de IAM, elija Entidad principal de IAM. En Tipo de entidad principal de IAM, elija Usuario o Rol. A continuación, en Usuario de entidad principal de IAM, elija una opción de la lista o introduzca el ID de la identidad.

c. Para crear la concesión de S3 Access Grants, seleccione Siguiente o Crear concesión.

4. Si Siguiente o Crear autorización están desactivadas:

No se puede crear la concesión

- Es posible que primero tenga que registrar una ubicación en su instancia de S3 Access Grants.
- Puede que no tenga el permiso `s3:CreateAccessGrant` para crear una concesión de acceso. Póngase en contacto con el administrador de la cuenta.

Uso de la AWS CLI

Para instalar AWS CLI, consulte Instalación de AWS CLI en la Guía del usuario de AWS Command Line Interface.

En los siguientes ejemplos se muestra cómo crear una solicitud de concesión de acceso para una entidad principal de IAM y cómo crear una solicitud de concesión de acceso para un usuario o grupo del directorio corporativo.

Para utilizar los comandos de ejemplo siguientes, sustituya *`user input placeholders`* con su información.

> **Note**
>
> Si va a crear una concesión de acceso que otorgue acceso a un solo objeto, incluya el parámetro `--s3-prefix-type Object` necesario.

Example Creación de una solicitud de concesión de acceso para una entidad principal de 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 Creación de una respuesta a la concesión de acceso

```
{"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"
}
```

Creación de una solicitud de concesión de acceso para un usuario o grupo de directorio

Para crear una solicitud de concesión de acceso para un usuario o grupo del directorio, primero debe obtener el GUID del usuario o grupo del directorio mediante la ejecución de uno de los siguientes comandos.

Example Obtener un GUID para un usuario o un grupo de directorio

Puede encontrar el GUID de un usuario del Centro de identidades de IAM a través de la consola del Centro de identidades de IAM o mediante la AWS CLI o los SDK de AWS. El siguiente comando muestra los usuarios de la instancia de Centro de identidades de IAM especificada, con sus nombres e identificadores.

```
aws identitystore list-users --identity-store-id d-1a2b3c4d1234
```

Este comando contiene una lista de los grupos de la instancia del Centro de identidades de IAM especificada.

```
aws identitystore list-groups --identity-store-id d-1a2b3c4d1234
```

Example Crear una concesión de acceso para un usuario o grupo de directorio

Este comando es similar a la creación de una concesión para los usuarios o roles de IAM, excepto que el tipo de beneficiario es `DIRECTORY_USER` o `DIRECTORY_GROUP` y el identificador del beneficiario es el GUID del usuario o grupo del directorio.

```
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 \
```

Uso de la API de REST

Para obtener información sobre la compatibilidad con la API de REST de Amazon S3 para administrar concesiones de acceso, consulte las siguientes secciones en la Referencia de la API de Amazon Simple Storage Service:

- CreateAccessGrant
- DeleteAccessGrant
- GetAccessGrant
- ListAccessGrants

Uso de los AWS SDK

En esta sección se proporcionan ejemplos de cómo crear una concesión de acceso mediante los SDK de AWS.

Java

> Para utilizar el ejemplo siguiente, sustituya *user input placeholders* con su propia información:

> **Note**
>
> Si va a crear una concesión de acceso que otorgue acceso a un solo objeto, incluya el parámetro `.s3PrefixType(S3PrefixType.Object)` necesario.

Example Creación de una solicitud de concesión de acceso

```
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 Creación de una respuesta a la concesión de acceso

```
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
)
```

Temas

Visualización de una concesión

Puede ver los detalles de una concesión de acceso en su instancia de Amazon S3 Access Grants mediante la consola de Amazon S3, la AWS Command Line Interface (AWS CLI), la API de REST de Amazon S3 y los SDK de AWS.

Uso de la consola de S3

Para ver los detalles de una concesión de acceso

1. Inicie sesión AWS Management Console Management Console y abra la consola de Amazon S3 en https://console.aws.amazon.com/s3/.
2. En el panel de navegación izquierdo, seleccione Concesiones de acceso.
3. En la página S3 Access Grants, seleccione la región que contiene la instancia de S3 Access Grants con la que quiere trabajar.
4. Seleccione Ver detalles de la instancia.
5. En la página de detalles, elija la pestaña Concesiones.
6. En la sección Concesiones, busque la concesión de acceso que desea ver. Utilice el cuadro de búsqueda para utilizar el cuadro de concesiones.

Uso de la AWS CLI

Para instalar AWS CLI, consulte Instalación de AWS CLI en la Guía del usuario de AWS Command Line Interface.

Para utilizar los comandos de ejemplo siguientes, sustituya *`user input placeholders`* con su información.

Example – Obtener los detalles de una concesión de acceso

```
aws s3control get-access-grant \
--account-id 111122223333 \
```

```
--access-grant-id a1b2c3d4-5678-90ab-cdef-EXAMPLE22222
```

Respuesta:

```
{
    "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://amzn-s3-demo-bucket/"
}
```

Example – Enumerar todas las concesiones de acceso en una instancia de S3 Access Grants

Si lo desea, puede utilizar los siguientes parámetros para restringir los resultados a un prefijo o identidad de AWS Identity and Access Management (IAM) de S3:

- Subprefijo: `--grant-scope s3://bucket-name/prefix*`
- Identidad de IAM: `--grantee-type IAM` y `--grantee-identifier arn:aws:iam::123456789000:role/accessGrantsConsumerRole`

```
aws s3control list-access-grants \
--account-id 111122223333
```

Respuesta:

```
{
    "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://amzn-s3-demo-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://amzn-s3-demo-bucket/prefixB*"
        },

    ]
}
```

Uso de la API de REST

Puede utilizar las operaciones de la API de Amazon S3 para ver los detalles de una concesión de acceso y enumerar todas las concesiones de acceso de una instancia de S3 Access Grants. Para obtener información sobre la compatibilidad con la API de REST para administrar concesiones de acceso, consulte las siguientes secciones en la Referencia de la API de Amazon Simple Storage Service:

- GetAccessGrant
- ListAccessGrants

Uso de los AWS SDK

En esta sección se proporcionan ejemplos de cómo obtener los detalles de una concesión de acceso con los SDK de AWS.

Para utilizar los siguientes ejemplos, reemplace los *user input placeholders* con su propia información.

Java

Example – Obtener los detalles de una concesión de acceso

```
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);
}
```

Respuesta:

```
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://amzn-s3-demo-bucket/
)
```

Example – Enumerar todas las concesiones de acceso en una instancia de S3 Access Grants

Si lo desea, puede utilizar estos parámetros para restringir los resultados a un prefijo o identidad de IAM de S3:

- Alcance: `GrantScope=s3://bucket-name/prefix*`
- Beneficiario: `GranteeType=IAM` y `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);
}
```

Respuesta:

```
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://amzn-s3-demo-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://amzn-s3-demo-bucket/prefixB*
)
]
)
```

Eliminar una concesión

Puede eliminar las concesiones de acceso de su instancia de Amazon S3 Access Grants. No se puede deshacer la eliminación de una concesión de acceso. Tras eliminar una concesión de acceso, el beneficiario dejará de tener acceso a sus datos de Amazon S3.

Puede eliminar una concesión de acceso mediante la consola de Amazon S3, la AWS Command Line Interface (AWS CLI), la API de REST de Amazon S3 y los SDK de AWS.

Uso de la consola de S3

Para eliminar una concesión de acceso

1. Inicie sesión AWS Management Console Management Console y abra la consola de Amazon S3 en https://console.aws.amazon.com/s3/.
2. En el panel de navegación izquierdo, seleccione Concesiones de acceso.
3. En la página S3 Access Grants, seleccione la región que contiene la instancia de S3 Access Grants con la que quiere trabajar.
4. Seleccione Ver detalles de la instancia.
5. En la página de detalles, elija la pestaña Concesiones.
6. Busque la concesión que desea eliminar. Cuando encuentre la concesión, seleccione el botón de opción situado junto a ella.
7. Elija Eliminar. Aparece un cuadro de diálogo con una advertencia de que la acción no se puede deshacer. Vuelva a seleccionar Eliminar para eliminar la concesión.

Uso de la AWS CLI

Para instalar AWS CLI, consulte Instalación de AWS CLI en la Guía del usuario de AWS Command Line Interface.

Para utilizar el comando de ejemplo siguiente, sustituya *user input placeholders* con su información.

Example – Eliminar una concesión de acceso

```
aws s3control delete-access-grant \
--account-id 111122223333 \
--access-grant-id a1b2c3d4-5678-90ab-cdef-EXAMPLE11111
```

```
// No response body
```

Uso de la API de REST

Para obtener información sobre la compatibilidad con la API de REST de Amazon S3 para administrar concesiones de acceso, consulte DeleteAccessGrant en la Referencia de la API de Amazon Simple Storage Service.

Uso de los AWS SDK

En esta sección se proporcionan ejemplos de cómo eliminar una concesión de acceso mediante los SDK de AWS. Para utilizar el ejemplo siguiente, sustituya *user input placeholders* con su propia información.

Java

Example – Eliminar una concesión de acceso

```
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);
}
```

Respuesta:

```
DeleteAccessGrantResponse()
```

Solicitar acceso a los datos de Amazon S3 a través de S3 Access Grants

Después de usar Concesiones de acceso Amazon S3 para crear una concesión de acceso que otorgue a las entidades principales de AWS Identity and Access Management (IAM), las identidades de su directorio corporativo o las aplicaciones autorizadas acceso a sus datos de S3, los beneficiarios pueden solicitar credenciales para acceder a estos datos.

Cuando una aplicación o Servicio de AWS utiliza la operación de la API `GetDataAccess` para pedir acceso a S3 Access Grants a sus datos de S3 en nombre de un beneficiario, S3 Access

Grants comprueba primero que ha concedido acceso a los datos a esta identidad. A continuación, S3 Access Grants utiliza la operación de la API AssumeRole para obtener un token de credencial temporal y se lo ofrece al solicitante. Este token de credencial temporal es un token de AWS Security Token Service token (AWS STS).

La solicitud `GetDataAccess` debe incluir el parámetro `target`, que especifica el alcance de los datos de S3 a los que se aplican las credenciales temporales. Este alcance `target` puede ser el mismo que el alcance de la concesión o un subconjunto del mismo, pero el alcance `target` debe estar dentro del alcance de la concesión que se otorgó al solicitante. La solicitud también debe especificar el parámetro `permission` para indicar el nivel de permiso para las credenciales temporales, ya sea `READ`, `WRITE` o `READWRITE`.

El solicitante puede especificar el nivel de privilegio del token temporal en su solicitud de credenciales. Con el parámetro `privilege`, el solicitante puede reducir o aumentar el alcance de acceso de las credenciales temporales, dentro de los límites del alcance de la concesión. El valor predeterminado del parámetro `privilege` es `Default`, lo que significa que el alcance objetivo de la credencial devuelta es el alcance de la concesión original. El otro valor posible para `privilege` es `Minimal`. Si el alcance `target` se reduce con respecto al alcance de la concesión original, se elimina el alcance de la credencial temporal para que coincida con el alcance `target`, siempre que el alcance `target` esté dentro del alcance de la concesión.

En la siguiente tabla se detalla el efecto del parámetro `privilege` en dos concesiones. Una de las concesiones tiene el alcance `S3://amzn-s3-demo-bucket1/bob/*`, que incluye todo el prefijo `bob/` del bucket *amzn-s3-demo-bucket1*. La otra concesión tiene el alcance `S3://amzn-s3-demo-bucket1/bob/reports/*`, que incluye solo el prefijo `bob/reports/` del bucket *amzn-s3-demo-bucket1*.

Alcance de la concesión	Alcance solicitado	Privilegio	Alcance devuelto	Efecto
`S3://amzn-s3-demo-bucket1 /bob/*`	`amzn-s3-demo-bucket1 /bob/*`	`Default`	*amzn-s3-demo-bucket1* `/bob/*`	El solicitante tiene acceso a todos los objetos que tienen nombres clave que comienzan con el prefijo *bob/* del bucket *amzn-s3-demo-bucket1* .

Alcance de la concesión	Alcance solicitado	Privilegio	Alcance devuelto	Efecto
S3://*amzn-s3-demo-bucket1* /bob/*	*amzn-s3-demo-bucket1* /bob/	Minimal	*amzn-s3-demo-bucket1* /bob/	Sin el carácter comodín * después del nombre del prefijo bob/, el solicitante solo tiene acceso al objeto denominado bob/ en el bucket *amzn-s3-demo-bucket1* . No es habitual tener un objeto de este tipo. El solicitante no tiene acceso a ningún otro objeto, incluidos los que tienen nombres clave que comiencen con el prefijo bob/.
S3://*amzn-s3-demo-bucket1* /bob/*	*amzn-s3-demo-bucket1* /bob/images/*	Minimal	*amzn-s3-demo-bucket1* /bob/images/*	El solicitante tiene acceso a todos los objetos que tienen nombres clave que comienzan con el prefijo *bob/images/** del bucket *amzn-s3-demo-bucket1* .
S3://*amzn-s3-demo-bucket1* /bob/reports/*	*amzn-s3-demo-bucket1* /bob/reports/file.txt	Default	*amzn-s3-demo-bucket1* /bob/reports/*	El solicitante tiene acceso a todos los objetos que tienen nombres clave que comienzan con el prefijo bob/reports del bucket *amzn-s3-demo-bucket1* , que es el alcance de la concesión coincidente.

Alcance de la concesión	Alcance solicitado	Privilegio	Alcance devuelto	Efecto
S3://*amzn-s3-demo-bucket1* /bob/reports/*	*amzn-s3-demo-bucket1* /bob/reports/file.txt	Minimal	*amzn-s3-demo-bucket1* /bob/reports/file.txt	El solicitante solo tiene acceso al objeto con el nombre de clave bob/reports/file.txt del bucket *amzn-s3-demo-bucket1* . El solicitante no tiene acceso a ningún otro objeto.

El parámetro `durationSeconds` establece la duración de la credencial temporal, en segundos. El valor predeterminado es `3600` segundos (1 hora), pero el solicitante (el beneficiario) puede especificar un intervalo de `900` segundos (15 minutos) a `43200` segundos (12 horas). Si el beneficiario solicita un valor superior al indicado en este máximo, la solicitud producirá un error.

> **Note**
>
> En su solicitud de un token temporal, si la ubicación es un objeto, establece el valor del parámetro `targetType` de su solicitud en `Object`. Este parámetro solo es necesario si la ubicación es un objeto y el nivel de privilegio es `Minimal`. Si la ubicación es un bucket o un prefijo, no es necesario que especifique este parámetro.

Para obtener más información, consulte GetDataAccess en la Referencia de la API de Amazon Simple Storage Service.

Puede solicitar credenciales temporales mediante AWS Command Line Interface (AWS CLI), la API de REST de Amazon S3 y los SDK de AWS.

Uso de la AWS CLI

Para instalar AWS CLI, consulte Instalación de AWS CLI en la Guía del usuario de AWS Command Line Interface.

Para utilizar el comando de ejemplo siguiente, sustituya *user input placeholders* con su información.

Example Solicitar credenciales temporales

Solicitud:

```
aws s3control get-data-access \
--account-id 111122223333 \
--target s3://amzn-s3-demo-bucket/prefixA* \
--permission READ \
--privilege Default \
--region us-east-2
```

Response: (Respuesta:)

```
{
"Credentials": {
 "AccessKeyId": "Example-key-id",
 "SecretAccessKey": "Example-access-key",
 "SessionToken": "Example-session-token",
"Expiration": "2023-06-14T18:56:45+00:00"},
    "MatchedGrantTarget": "s3://amzn-s3-demo-bucket/prefixA**"
}
```

Uso de la API de REST

Para obtener información sobre la compatibilidad con la API de REST de Amazon S3 para solicitar credenciales temporales desde S3 Access Grants, consulte GetDataAccess en la Referencia de la API de Amazon Simple Storage Service.

Uso de los AWS SDK

En esta sección se proporciona un ejemplo de cómo los beneficiarios solicitan credenciales temporales a S3 Access Grants mediante los SDK de AWS.

Java

> En el siguiente ejemplo de código se devuelven las credenciales temporales que el beneficiario utiliza para acceder a sus datos de S3. Para utilizar este ejemplo de código, reemplace *user input placeholders* con su propia información.

Example Obtener credenciales temporales

Solicitud:

```
public void getDataAccess() {
GetDataAccessRequest getDataAccessRequest = GetDataAccessRequest.builder()
.accountId("111122223333")
.permission(Permission.READ)
.privilege(Privilege.MINIMAL)
.target("s3://amzn-s3-demo-bucket/prefixA*")
.build();
GetDataAccessResponse getDataAccessResponse =
 s3Control.getDataAccess(getDataAccessRequest);
LOGGER.info("GetDataAccessResponse: " + getDataAccessResponse);
}
```

Response: (Respuesta:)

```
GetDataAccessResponse(
Credentials=Credentials(
AccessKeyId="Example-access-key-id",
SecretAccessKey="Example-secret-access-key",
SessionToken="Example-session-token",
Expiration=2023-06-07T06:55:24Z
))
```

Acceder a los datos de S3 mediante una concesión de acceso

Una vez que el beneficiario obtiene las credenciales temporales mediante su concesión de acceso, puede utilizarlas para llamar a las operaciones de la API de Amazon S3 para acceder a sus datos.

Los beneficiarios pueden obtener acceso a los datos de S3 a través de AWS Command Line Interface (AWS CLI), los SDK de AWS y la API de REST de Amazon S3.

Uso de la AWS CLI

Una vez que el beneficiario obtiene sus credenciales temporales de S3 Access Grants, puede configurar un perfil con estas credenciales para recuperar los datos.

Para instalar AWS CLI, consulte Instalación de AWS CLI en la Guía del usuario de AWS Command Line Interface.

Para utilizar los comandos de ejemplo siguientes, sustituya *user input placeholders* con su información.

Example — Configurar un perfil

```
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
```

Para utilizar el comando de ejemplo siguiente, sustituya *user input placeholders* con su información.

Example – Obtener los datos de S3

El beneficiario puede usar el comando get-objectAWS CLI para acceder a los datos. El beneficiario también puede usar put-object, ls y otros comandos de AWS CLI de S3.

```
aws s3api get-object \
--bucket amzn-s3-demo-bucket1 \
--key myprefix \
--region us-east-2 \
--profile access-grants-consumer-access-profile
```

Uso de los AWS SDK

En esta sección se proporcionan ejemplos de cómo los beneficiarios pueden acceder a sus datos de S3 mediante los SDK de AWS.

Java

> Para ver ejemplos de cómo obtener datos de S3 mediante credenciales temporales, consulte cómo obtener un objeto mediante los SDK de AWS y ejemplos de código de Amazon S3 para AWS SDK for Java 2.x.

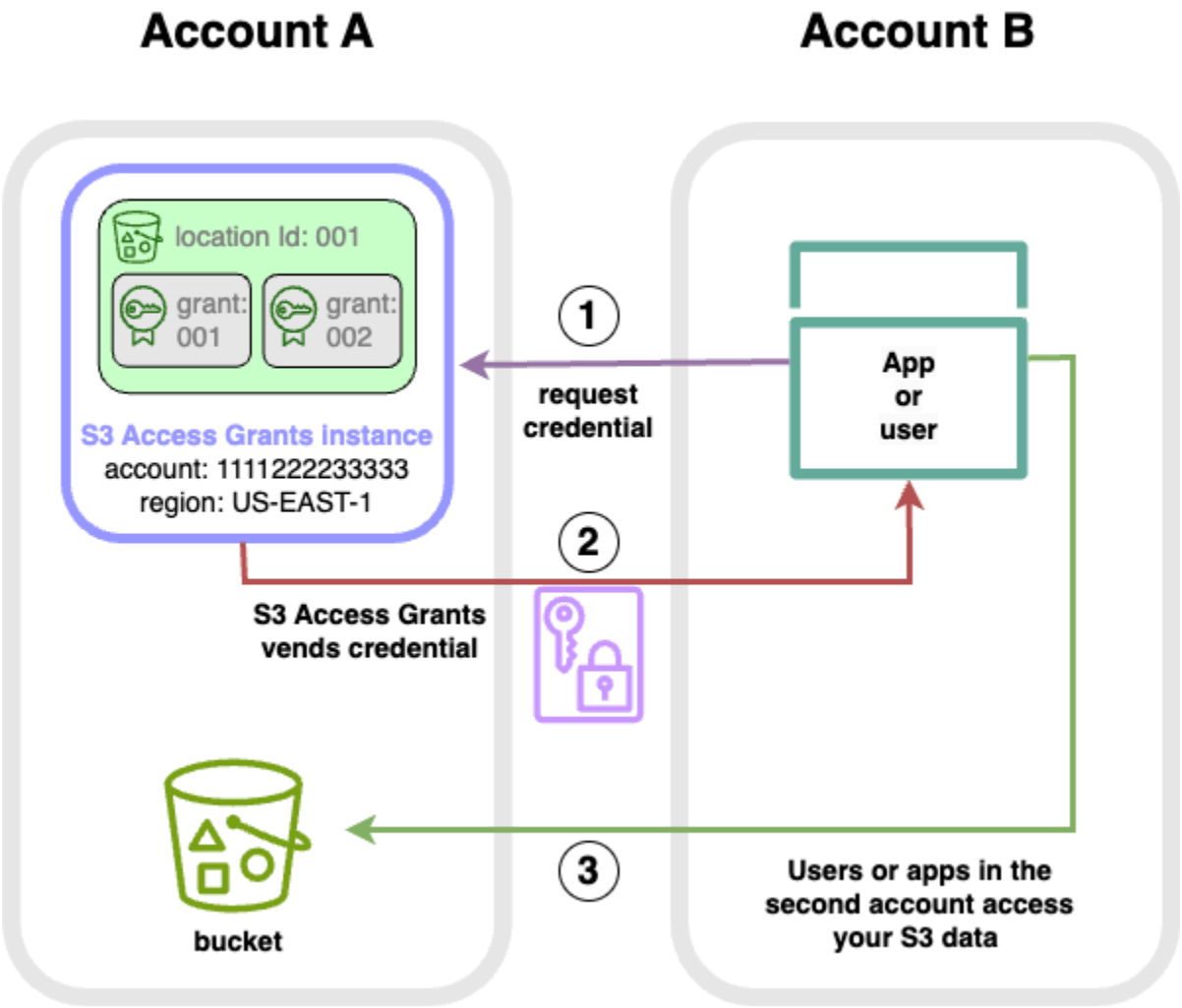
Acceso entre cuentas a S3 Access Grants

Con S3 Access Grants, puede conceder acceso a los datos de Amazon S3 a lo siguiente:

- Identidades de AWS Identity and Access Management (IAM) de su cuenta
- Identidades de IAM en otras cuentas de AWS
- Usarios o grupos de directorios en su instancia de AWS IAM Identity Center

En primer lugar, configure el acceso entre cuentas para la otra cuenta. Esto incluye conceder acceso a su instancia de S3 Access Grants mediante una política de recursos. A continuación, conceda acceso a sus datos de S3 (buckets, prefijos u objetos) mediante concesiones.

Tras configurar el acceso entre cuentas, la otra cuenta puede solicitar credenciales de acceso temporales a sus datos de Amazon S3 desde S3 Access Grants. La siguiente imagen muestra el flujo de usuarios para el acceso de S3 entre cuentas a través de S3 Access Grants:



1. Los usuarios o las aplicaciones de una segunda cuenta (B) solicitan credenciales a la instancia de S3 Access Grants de su cuenta (A), donde se almacenan los datos de Amazon S3. Para obtener

más información, consulte Solicitar acceso a los datos de Amazon S3 a través de S3 Access Grants.

2. La instancia de S3 Access Grants de su cuenta (A) devuelve credenciales temporales si hay una concesión que permite a la segunda cuenta acceder a sus datos de Amazon S3. Para obtener más información, consulte the section called “Crear concesiones”.
3. Los usuarios o las aplicaciones de la segunda cuenta (B) utilizan las credenciales ofrecidas por S3 Access Grants para acceder a los datos de S3 de su cuenta (A).

Configuración del acceso entre cuentas a S3 Access Grants

Para conceder acceso a S3 entre cuentas a través de S3 Access Grants, siga estos pasos:

- Paso 1: configure una instancia de S3 Access Grants en su cuenta, por ejemplo, el ID de la cuenta `111122223333`, donde se almacenan los datos de S3.
- Paso 2: configure la política de recursos de la instancia de S3 Access Grants de su cuenta `111122223333` para dar acceso a la segunda cuenta, por ejemplo, el ID de cuenta `444455556666`.
- Paso 3: configure los permisos de IAM para que la entidad principal de IAM de la segunda cuenta `444455556666` solicite las credenciales de la instancia de S3 Access Grants de su cuenta `111122223333`.
- Paso 4: cree una concesión en su cuenta `111122223333` que permita a la entidad principal de IAM de la segunda cuenta `444455556666` acceder a algunos de los datos de S3 de su cuenta `111122223333`.

Paso 1: configuración de una instancia de S3 Access Grants en su cuenta

En primer lugar, debe tener una instancia de S3 Access Grants en su cuenta `111122223333` para administrar el acceso a sus datos de Amazon S3. Debe crear una instancia de S3 Access Grants en cada una de las Región de AWS en las que se almacenen los datos de S3 que desea compartir. Si comparte datos en más de una Región de AWS, repita cada uno de estos pasos de configuración para cada Región de AWS. Si ya tiene una instancia de S3 Access Grants en la Región de AWS en la que se almacenen sus datos de S3, vaya al siguiente paso. Si no ha configurado una instancia de S3 Access Grants, consulte Crear una instancia de S3 Access Grants para completar este paso.

Paso 2: Configure la política de recursos de su instancia de S3 Access Grants para conceder acceso entre cuentas

Después de crear una instancia de S3 Access Grants en su cuenta `111122223333` para el acceso entre cuentas, configure la política basada en recursos para la instancia de S3 Access Grants de su cuenta `111122223333`, con el a fin de conceder el acceso entre cuentas. La instancia de S3 Access Grants admite políticas basadas en recursos. Con la política adecuada basada en los recursos, es posible conceder acceso a usuarios o roles de AWS Identity and Access Management (IAM) de otras Cuentas de AWS a su instancia de S3 Access Grants. El acceso entre cuentas solo concede los siguientes permisos (acciones):

- `s3:GetAccessGrantsInstanceForPrefix`: el usuario, el rol o la app pueden recuperar la instancia de S3 Access Grants que contiene un prefijo concreto.
- `s3:ListAccessGrants`
- `s3:ListAccessLocations`
- `s3:GetDataAccess`: el usuario, el rol o la app pueden solicitar credenciales temporales en función del acceso que se le concedió a través de S3 Access Grants. Utilice estas credenciales para acceder a los datos de S3 a los que se le ha concedido acceso.

Puede elegir cuáles de estos permisos desea incluir en la política de recursos. Esta política de recursos de la instancia de S3 Access Grants es una política normal basada en recursos y es compatible con todo lo que admite el lenguaje de la política de IAM. En la misma política, puede conceder acceso a identidades de IAM específicas en la cuenta `111122223333`, por ejemplo, mediante la condición `aws:PrincipalArn`, pero no es necesario hacerlo con S3 Access Grants. En su lugar, en su instancia de S3 Access Grants, puede crear concesiones para identidades de IAM individuales de su cuenta, así como para la otra cuenta. Al administrar cada concesión de acceso a través de S3 Access Grants, puede escalar sus permisos.

Si ya usa AWS Resource Access Manager (AWS RAM), puede utilizarlo para compartir sus recursos de `s3:AccessGrants` con otras cuentas o dentro de su organización. Para obtener más información, consulte Trabajo con recursos compartidos de AWS. Si no utiliza AWS RAM, también puede añadir la política de recursos mediante las operaciones de la API de S3 Access Grants o la AWS Command Line Interface (AWS CLI).

Uso de la consola de S3

Recomendamos que se use la consola de AWS Resource Access Manager (AWS RAM) para compartir sus recursos de `s3:AccessGrants` con otras cuentas o dentro de su organización. Para compartir entre cuentas de S3 Access Grants, haga lo siguiente:

Configuración de la política de recursos de la instancia de S3 Access Grants:

1. Inicie sesión AWS Management Console Management Console y abra la consola de Amazon S3 en https://console.aws.amazon.com/s3/.
2. Seleccione la Región de AWS en el selector de Región de AWS.
3. En el panel de navegación izquierdo, seleccione Access Grants (Concesiones de acceso).
4. En la página de instancias de Access Grants, en la sección Instancia de esta cuenta, seleccione Compartir instancia. Esto le redirige a la consola de AWS RAM.
5. Seleccione Crear recurso compartido.
6. Siga los pasos de AWS RAM para crear un recurso compartido. Para obtener más información, consulte Crear un recurso compartido en AWS RAM.

Uso de la AWS CLI

Para instalar AWS CLI, consulte Instalación de AWS CLI en la Guía del usuario de AWS Command Line Interface.

Puede agregar la política de recursos mediante el comando `put-access-grants-instance-resource-policy` de la CLI.

Si desea conceder acceso entre cuentas a la instancia de S3 Access Grants que se encuentra en su cuenta `111122223333` a la segunda cuenta `444455556666`, la política de recursos de la instancia de S3 Access Grants de su cuenta `111122223333` debe dar permiso a la segunda cuenta `444455556666` para realizar las siguientes acciones:

- `s3:ListAccessGrants`
- `s3:ListAccessGrantsLocations`
- `s3:GetDataAccess`
- `s3:GetAccessGrantsInstanceForPrefix`

En la política de recursos de la instancia de S3 Access Grants, especifique el ARN de su instancia de S3 Access Grants como `Resource` y la segunda cuenta `444455556666` como `Principal`. En el siguiente ejemplo, sustituya los *marcadores de posición del usuario* con su propia información.

```
{
"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"
} ]
}
```

Para añadir o actualizar una política de recursos de S3 Access Grants, puede usar el siguiente comando. Al utilizar el comando de ejemplo siguiente, sustituya *user input placeholders* con su información.

Example Añadir o actualizar una política de recursos de instancia de 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 Obtener una política de recursos de S3 Access Grants

También puede utilizar la CLI para obtener o eliminar una política de recursos para una instancia de S3 Access Grants.

Para obtener una política de recursos de S3 Access Grants, use el siguiente comando de ejemplo. Para utilizar este comando de ejemplo, sustituya *user input placeholders* por su propia información.

```
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 Eliminar una política de recursos de S3 Access Grants

Para eliminar una política de recursos de S3 Access Grants, use el siguiente comando de ejemplo. Para utilizar este comando de ejemplo, sustituya *user input placeholders* por su propia información.

```
aws s3control delete-access-grants-instance-resource-policy \
--account-id 111122223333 \
```

```
--region us-east-2

// No response body
```

Uso de la API de REST

Puede agregar la política de recursos mediante la API PutAccessGrantsInstanceResourcePolicy.

Si desea conceder acceso entre cuentas a la instancia de S3 Access Grants que se encuentra en su cuenta 111122223333 a la segunda cuenta 444455556666, la política de recursos de la instancia de S3 Access Grants de su cuenta 111122223333 debe dar permiso a la segunda cuenta 444455556666 para realizar las siguientes acciones:

- `s3:ListAccessGrants`
- `s3:ListAccessGrantsLocations`
- `s3:GetDataAccess`
- `s3:GetAccessGrantsInstanceForPrefix`

En la política de recursos de la instancia de S3 Access Grants, especifique el ARN de su instancia de S3 Access Grants como `Resource` y la segunda cuenta 444455556666 como `Principal`. Para utilizar el siguiente ejemplo, sustituya los *marcadores de posición del usuario* con su propia información.

```
{
"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"
} ]
}
```

A continuación, puede usar la API PutAccessGrantsInstanceResourcePolicy para configurar la política

Para obtener información sobre la compatibilidad con la API de REST para actualizar, obtener o eliminar una política de recursos para una instancia de S3 Access Grants, consulte las siguientes secciones en la Referencia de la API de Amazon Simple Storage Service:

- PutAccessGrantsInstanceResourcePolicy
- GetAccessGrantsInstanceResourcePolicy
- DeleteAccessGrantsInstanceResourcePolicy

Uso de los AWS SDK

En esta sección, encontrará ejemplos de SDK de AWS sobre cómo configurar su política de recursos de S3 Access Grants para conceder a una segunda cuenta de AWS acceso a algunos de sus datos de S3.

Java

Añada, actualice, obtenga o elimine una política de recursos para administrar el acceso entre cuentas a su instancia de S3 Access Grants.

Example Añadir o actualizar una política de recursos de instancia de S3 Access Grants

Si desea conceder acceso entre cuentas a la instancia de S3 Access Grants que se encuentra en su cuenta `111122223333` a la segunda cuenta `444455556666`, la política de recursos de la instancia de S3 Access Grants de su cuenta `111122223333` debe dar permiso a la segunda cuenta `444455556666` para realizar las siguientes acciones:

- `s3:ListAccessGrants`
- `s3:ListAccessGrantsLocations`
- `s3:GetDataAccess`
- `s3:GetAccessGrantsInstanceForPrefix`

En la política de recursos de la instancia de S3 Access Grants, especifique el ARN de su instancia de S3 Access Grants como `Resource` y la segunda cuenta `444455556666` como `Principal`. Para utilizar el siguiente ejemplo, sustituya los *`marcadores de posición del usuario`* con su propia información.

```
{
"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"
} ]
}
```

Para añadir o actualizar una política de recursos de instancia de S3 Access Grants, use el siguiente código de ejemplo:

```
public void putAccessGrantsInstanceResourcePolicy() {
 PutAccessGrantsInstanceResourcePolicyRequest putRequest =
 PutAccessGrantsInstanceResourcePolicyRequest.builder()
 .accountId(111122223333)
 .policy(RESOURCE_POLICY)
 .build();
 PutAccessGrantsInstanceResourcePolicyResponse putResponse =
 s3Control.putAccessGrantsInstanceResourcePolicy(putRequest);
 LOGGER.info("PutAccessGrantsInstanceResourcePolicyResponse: " + putResponse);
 }
```

Respuesta:

```
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 Obtener una política de recursos de S3 Access Grants

Para obtener una política de recursos de S3 Access Grants, use el siguiente código de ejemplo. Para utilizar el comando de ejemplo siguiente, sustituya *user input placeholders* con su información.

```
public void getAccessGrantsInstanceResourcePolicy() {
 GetAccessGrantsInstanceResourcePolicyRequest getRequest =
 GetAccessGrantsInstanceResourcePolicyRequest.builder()
 .accountId(111122223333)
 .build();
 GetAccessGrantsInstanceResourcePolicyResponse getResponse =
 s3Control.getAccessGrantsInstanceResourcePolicy(getRequest);
 LOGGER.info("GetAccessGrantsInstanceResourcePolicyResponse: " + getResponse);
 }
```

Respuesta:

```
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 Eliminar una política de recursos de S3 Access Grants

Para eliminar una política de recursos de S3 Access Grants, use el siguiente código de ejemplo. Para utilizar el comando de ejemplo siguiente, sustituya *user input placeholders* con su información.

```
public void deleteAccessGrantsInstanceResourcePolicy() {
 DeleteAccessGrantsInstanceResourcePolicyRequest deleteRequest =
 DeleteAccessGrantsInstanceResourcePolicyRequest.builder()
 .accountId(111122223333)
 .build();
 DeleteAccessGrantsInstanceResourcePolicyResponse deleteResponse =
 s3Control.putAccessGrantsInstanceResourcePolicy(deleteRequest);
 LOGGER.info("DeleteAccessGrantsInstanceResourcePolicyResponse: " + deleteResponse);
 }
```

Respuesta:

```
DeleteAccessGrantsInstanceResourcePolicyResponse()
```

Paso 3: concesión de permiso a las identidades de IAM de una segunda cuenta para llamar a la instancia de S3 Access Grants de su cuenta

Una vez que el propietario de los datos de Amazon S3 haya configurado la política multicuenta para la instancia de S3 Access Grants de la cuenta 111122223333, el propietario de la segunda cuenta 444455556666 debe crear una política basada en identidades para sus usuarios o funciones de IAM y el propietario debe darles acceso a la instancia de S3 Access Grants. En la política basada en identidades, incluya una o más de las siguientes acciones, en función de lo que se conceda en la política de recursos de instancia de S3 Access Grants y de los permisos que quiera conceder:

- `s3:ListAccessGrants`
- `s3:ListAccessGrantsLocations`
- `s3:GetDataAccess`
- `s3:GetAccessGrantsInstanceForPrefix`

Siguiendo el patrón de acceso entre cuentas de AWS, los usuarios o roles de IAM de la segunda cuenta 444455556666 deben tener de forma explícita uno o varios de estos permisos. Por ejemplo, conceda el permiso `s3:GetDataAccess` para que el usuario o rol de IAM pueda llamar a la instancia de S3 Access Grants de la cuenta 111122223333 para solicitar credenciales.

Para utilizar este comando de ejemplo, sustituya *`user input placeholders`* por su propia información.

```
{
```

```
 "Version": "2012-10-17",
 "Statement": [
 {
  "Effect": "Allow",
  "Action": [
   "s3:GetDataAccess",
  ],
   "Resource": "arn:aws:s3:us-east-2:111122223333:access-grants/default"
  }
 ]
}
```

Para obtener más información acerca de la edición de la política de IAM basada en identidades, consulte Edición de políticas de IAM en la Guía de AWS Identity and Access Management.

Paso 4: Cree una concesión en la instancia de S3 Access Grants de su cuenta que dé a la identidad IAM de la segunda cuenta acceso a algunos de los datos de S3

Para el paso de configuración final, puede crear una concesión en la instancia de S3 Access Grants de su cuenta 111122223333 que dé a la identidad IAM de la segunda cuenta 444455556666 acceso a algunos de los datos de S3 de su cuenta. Puede hacerlo mediante la consola de Amazon S3, la CLI, la API y los SDK. Para obtener más información, consulte Crear concesiones.

En la concesión, especifique el ARN de AWS de la identidad de IAM de la segunda cuenta y especifique la ubicación de los datos de S3 (un bucket, un prefijo u objeto) a la que va a conceder acceso. Esta ubicación ya debe estar registrada en su instancia de S3 Access Grants. Para obtener más información, consulte Registrar una ubicación. Opcionalmente, puede especificar un subprefijo. Por ejemplo, si la ubicación a la que va a conceder el acceso es un bucket y desea limitar aún más el acceso a un objeto específico de ese bucket, introduzca el nombre de la clave del objeto en el campo `S3SubPrefix`. O bien, si quiere limitar el acceso a los objetos del bucket con nombres de clave que comiencen con un prefijo concreto, por ejemplo, `2024-03-research-results/`, introduzca `S3SubPrefix=2024-03-research-results/`.

El siguiente es un ejemplo de comando de CLI para crear una concesión de acceso para una identidad en la segunda cuenta. Para obtener más información, consulte Crear concesiones. Para utilizar este comando de ejemplo, sustituya *`user input placeholders`* por su propia información.

```
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
```

Tras configurar el acceso entre cuentas, el usuario o rol de la segunda cuenta puede hacer lo siguiente:

- Llama a `ListAccessGrantsInstances` para ver una lista de las instancias de S3 Access Grants que ha compartido con ella a través de AWS RAM. Para obtener más información, consulte Ver los detalles de una instancia de S3 Access Grants.
- Solicita credenciales temporales de S3 Access Grants. Para obtener información acerca de cómo realizar estas solicitudes, consulte Solicitar acceso a los datos de Amazon S3 a través de S3 Access Grants.

Uso de etiquetas de AWS con S3 Access Grants

Las etiquetas de Amazon S3 Access Grants tienen características similares a las etiquetas de objetos de Amazon S3. Cada etiqueta es un par clave-valor. Los recursos de S3 Access Grants que puede etiquetar son instancias, ubicaciones y concesiones de S3 Access Grants.

> **Note**
>
> El etiquetado en S3 Access Grants utiliza operaciones de API diferentes a las del etiquetado de objetos. S3 Access Grants utiliza las operaciones de la API TagResource, UntagResource y ListTagsForResource en las que un recurso puede ser una instancia de S3 Access Grants, una ubicación registrada o una concesión de acceso.

Al igual que las etiquetas de objetos, se aplican las siguientes limitaciones:

- Puede añadir etiquetas a los nuevos recursos de S3 Access Grants al crearlos, o puede añadir etiquetas a los recursos existentes.
- Puede asociar hasta 10 etiquetas a un recurso. Si hay varias etiquetas asociadas al mismo recurso, deben tener claves de etiquetas únicas.
- Una clave de etiqueta puede tener una longitud de hasta 128 caracteres Unicode y los valores de etiqueta pueden tener una longitud de hasta 256 caracteres Unicode. Las etiquetas se representan internamente en UTF-16. En UTF-16, los caracteres ocupan 1 o 2 posiciones de caracteres.

- Las claves y los valores distinguen entre mayúsculas y minúsculas.

Para obtener más información sobre las restricciones de las etiquetas, consulte Restricciones de las etiquetas definidas por el usuario en la Guía del usuario de AWS Billing.

Puede etiquetar los recursos en S3 Access Grants mediante la AWS Command Line Interface (AWS CLI), la API de REST de Amazon S3 o los SDK de AWS.

Uso de la AWS CLI

Para instalar AWS CLI, consulte Instalación de AWS CLI en la Guía del usuario de AWS Command Line Interface.

Puede etiquetar un recurso de S3 Access Grants al crearlo o bien después de crearlo. En los siguientes ejemplos se muestra cómo etiquetar o quitar etiquetas de una instancia de S3 Access Grants. Puede realizar operaciones similares para las ubicaciones registradas y las concesiones de acceso.

Para utilizar los comandos de ejemplo siguientes, sustituya *user input placeholders* con su información.

Example — Crear una instancia de S3 Access Grants con etiquetas

```
aws s3control create-access-grants-instance \
 --account-id 111122223333 \
 --profile access-grants-profile \
 --region us-east-2 \
 --tags Key=tagKey1,Value=tagValue1
```

Respuesta:

```
 {
    "CreatedAt": "2023-10-25T01:09:46.719000+00:00",
    "AccessGrantsInstanceId": "default",
    "AccessGrantsInstanceArn": "arn:aws:s3:us-east-2:111122223333:access-grants/
default"
}
```

Example — Etiquetar una instancia de S3 Access Grants ya creada

```
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 — Enumerar etiquetas de la instancia de S3 Access Grants

```
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
```

Respuesta:

```
{
    "Tags": [
        {
            "Key": "tagKey1",
            "Value": "tagValue1"
        },
        {
            "Key": "tagKey2",
            "Value": "tagValue2"
        }
    ]
}
```

Example — Desetiquetar la instancia de S3 Access Grants

```
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"
```

Uso de la API de REST

Puede usar la API de Amazon S3 para etiquetar, desetiquetar o mostrar las etiquetas de una instancia de S3 Access Grants, una ubicación registrada o una concesión de acceso. Para obtener

información sobre la compatibilidad con la API de REST para administrar etiquetas de S3 Access Grants, consulte las siguientes secciones en la Referencia de la API de Amazon Simple Storage Service:

- TagResource
- UntagResource
- ListTagsForResource

Limitaciones de S3 Access Grants

S3 Access Grants tiene las siguientes limitaciones:

> **Note**
>
> Si su caso de uso supera estas limitaciones, póngase en contacto con el servicio de asistencia de AWS para solicitar límites superiores.

Instancia de S3 Access Grants

Puede crear 1 instancia de S3 Access Grants por Región de AWS por cuenta. Consulte Crear una instancia de S3 Access Grants.

Ubicación de S3 Access Grants

Puede registrar 1000 ubicaciones de S3 Access Grants por instancia de S3 Access Grants. Consulte Registrar una ubicación de S3 Access Grants.

Concesión

Puede crear 100 000 concesiones por instancia de S3 Access Grants. Consulte Crear una concesión.

Integraciones de S3 Access Grants

S3 Access Grants se puede utilizar con los siguientes servicios y características de AWS. Esta página se actualizará a medida que haya nuevas integraciones disponibles.

Amazon Athena

Uso de grupos de trabajo de Athena habilitados para IAM Identity Center

Amazon EMR

- Lanzamiento de un clúster de Amazon EMR con S3 Access Grants

Amazon EMR en EKS

- Lanzamiento de un clúster de Amazon EMR o EKS con S3 Access Grants

Aplicaciones de Amazon EMR sin servidor

- Lanzamiento de una aplicación de Amazon EMR sin servidor con S3 Access Grants

AWS IAM Identity Center

- Propagación de identidad de confianza en aplicaciones

Amazon SageMaker Studio

- Las concesiones de acceso a Amazon S3 ahora se integran con Amazon SageMaker Studio

Marcos Python de código abierto

- Las concesiones de acceso a Amazon S3 ahora se integran con los marcos Python de código abierto

Administración de acceso con ACL

Las listas de control de acceso (ACL) son una de las opciones basadas en recursos que puede utilizar para administrar el acceso a los buckets y objetos. Puede utilizar las ACL para otorgar permisos básicos de lectura o escritura a otras Cuentas de AWS. Hay límites en la administración de permisos con ACL.

Por ejemplo, puede otorgar permisos solo a otras Cuentas de AWS, pero no a los usuarios de la cuenta. No puede otorgar permisos condicionales ni tampoco puede denegar permisos explícitamente. Las ACL son adecuadas para situaciones específicas. Por ejemplo, si el propietario de un bucket permite a otras Cuentas de AWS cargar objetos, los permisos para estos objetos solo los puede administrar la Cuenta de AWS que es propietaria del objeto con una ACL de objetos.

S3 Object Ownership es una configuración de bucket de Amazon S3 que puede usar para controlar la propiedad de los objetos que se cargan en el bucket y para activar o desactivar las ACL. De forma predeterminada, la propiedad de objetos se establece en la configuración impuesta por el propietario del bucket. Además, todas las ACL están deshabilitadas. Cuando las ACL están deshabilitadas, el

propietario del bucket posee todos los objetos del bucket y administra su acceso de forma exclusiva mediante políticas de administración de acceso.

La mayoría de los casos de uso modernos de Amazon S3 ya no requieren el uso de ACL. Le recomendamos desactivar las ACL, excepto en circunstancias inusuales en las que necesite controlar el acceso a cada objeto de manera individual. Si las ACL están desactivadas, puede usar políticas para controlar el acceso a todos los objetos del bucket, independientemente de quién haya subido los objetos al bucket. Para obtener más información, consulte Control de la propiedad de los objetos y desactivación de las ACL del bucket.

> **Important**
>
> Si el bucket utiliza la configuración de propietario del bucket obligatorio de S3 Object Ownership, debe utilizar políticas para conceder acceso al bucket y a los objetos que contiene. Si la configuración impuesta por el propietario del bucket está activada, las solicitudes de configuración o actualización de las listas de control de acceso (ACL) fallan y devuelven el código de error `AccessControlListNotSupported`. Las solicitudes de lectura de ACL siguen siendo compatibles.

Para obtener más información acerca de las ACL, vea los temas siguientes.

Temas

- Información general de las Listas de control de acceso (ACL)
- Configuración de la ACL
- Ejemplos de políticas para ACL

Información general de las Listas de control de acceso (ACL)

Las listas de control de acceso (ACL) de Amazon S3 le permiten administrar el acceso a buckets y objetos. Cada bucket y objeto incluye una ACL como un subrecurso. La ACL define qué Cuentas de AWS o grupos cuentan con acceso y el tipo de acceso que tienen. Cuando se recibe una solicitud en relación con un recurso, Amazon S3 verifica la ACL correspondiente para asegurarse de que el solicitante tenga los permisos de acceso necesarios.

S3 Object Ownership es una configuración de bucket de Amazon S3 que puede usar para controlar la propiedad de los objetos que se cargan en el bucket y para activar o desactivar las ACL. De forma

predeterminada, la propiedad de objetos se establece en la configuración impuesta por el propietario del bucket. Además, todas las ACL están deshabilitadas. Cuando las ACL están deshabilitadas, el propietario del bucket posee todos los objetos del bucket y administra su acceso de forma exclusiva mediante políticas de administración de acceso.

La mayoría de los casos de uso modernos de Amazon S3 ya no requieren el uso de ACL. Le recomendamos desactivar las ACL, excepto en circunstancias inusuales en las que necesite controlar el acceso a cada objeto de manera individual. Si las ACL están desactivadas, puede usar políticas para controlar el acceso a todos los objetos del bucket, independientemente de quién haya subido los objetos al bucket. Para obtener más información, consulte Control de la propiedad de los objetos y desactivación de las ACL del bucket.

> **Important**
>
> Si el bucket utiliza la configuración de propietario del bucket obligatorio de S3 Object Ownership, debe utilizar políticas para conceder acceso al bucket y a los objetos que contiene. Si la configuración impuesta por el propietario del bucket está activada, las solicitudes de configuración o actualización de las listas de control de acceso (ACL) fallan y devuelven el código de error `AccessControlListNotSupported`. Las solicitudes de lectura de ACL siguen siendo compatibles.

Cuando crea un bucket o un objeto, Amazon S3 crea una ACL predeterminada que concede al propietario del recurso control total sobre el recurso. Esto se muestra en el siguiente ACL de bucket de muestra (el ACL del objeto predeterminado tiene la misma estructura):

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>
```

La ACL de muestra incluye un elemento `Owner` que identifica al propietario a través del ID de usuario canónico de la cuenta de Cuenta de AWS. Para obtener instrucciones acerca de cómo buscar su identificador de usuario canónico, consulte Búsqueda del ID de usuario canónico de una Cuenta de AWS. El elemento `Grant` identifica al beneficiario (ya sea una Cuenta de AWS o un grupo predefinido) y el permiso otorgado. Esta ACL predeterminada tiene un elemento `Grant` para el propietario. Usted otorga permisos al añadir elementos `Grant`, con cada concesión identifica al beneficiario y al permiso.

> **Note**
>
> Una ACL puede tener hasta 100 concesiones.

Temas

- ¿Quién es un beneficiario?
- ¿Qué permisos puedo conceder?
- Valores de aclRequired para solicitudes comunes de Amazon S3
- ACL de muestra
- ACL predefinidas

¿Quién es un beneficiario?

El beneficiario puede ser una Cuenta de AWS o uno de los grupos predefinidos de Amazon S3. Puede otorgar permiso a una Cuenta de AWS con la dirección de correo electrónico o el ID de usuario canónico. Sin embargo, si proporciona una dirección de correo electrónico en su solicitud de concesión, Amazon S3 detecta el ID de usuario canónico para esa cuenta y lo añade a la ACL. Las ACL resultantes siempre incluyen el ID de usuario canónico de la Cuenta de AWS, no la dirección de correo electrónico de la Cuenta de AWS.

Cuando conceda derechos de acceso, especifique cada beneficiario como par *`type`*="*`value`*", donde *`type`* sea uno de los siguientes:

- `id`: si el valor especificado es el ID de usuario canónico de una Cuenta de AWS
- `uri`: si está otorgando permisos a un grupo predefinido
- `emailAddress`: si el valor especificado es la dirección de correo electrónico de una Cuenta de AWS

Important

Solo las siguientes regiones de AWS permiten utilizar direcciones de email para especificar el beneficiario:

- EE.UU. Este (Norte de Virginia)
- Oeste de EE. UU. (Norte de California)
- EE.UU. Oeste (Oregón)
- Asia Pacífico (Singapur)
- Asia-Pacífico (Sídney)
- Asia Pacífico (Tokio)
- Europa (Irlanda)
- América del Sur (São Paulo)

Para ver una lista de todas las regiones y puntos de conexión admitidos por Amazon S3, consulte Regiones y puntos de conexión en la Referencia general de Amazon Web Services.

Example Ejemplo: dirección de correo electrónico

Por ejemplo, el encabezado `x-amz-grant-read` siguiente otorga a las Cuentas de AWS identificadas por direcciones de correo electrónico permisos para leer los datos y los metadatos de los objetos:

```
x-amz-grant-read: emailAddress="xyz@example.com", emailAddress="abc@example.com"
```

Warning

Cuando otorgue a otras Cuentas de AWS acceso a sus recursos, tenga en cuenta que las Cuentas de AWS pueden delegar sus permisos a los usuarios de sus cuentas. Esto se

conoce como acceso entre cuentas. Para obtener información acerca del uso del acceso entre cuentas, consulte Creación de un rol para delegar permisos a un usuario de IAM en la guía del usuario de IAM.

Búsqueda del ID de usuario canónico de una Cuenta de AWS

El ID de usuario canónico está asociado a su Cuenta de AWS. Este ID es una cadena larga de caracteres, como:

```
79a59df900b949e55d96a1e698fbacedfd6e09d98eacf8f8d5218e7cd47ef2be
```

Para obtener información acerca de cómo encontrar un ID de usuario canónico para la cuenta, consulte Buscar el ID de usuario canónico de su Cuenta de AWS en la Guía de referencia de la Administración de cuentas de AWS.

También puede buscar el ID de usuario canónico de una Cuenta de AWS si lee la ACL de un bucket o un objeto para el cual la Cuenta de AWS tiene permisos de acceso. Cuando una solicitud de concesión otorga permisos a una Cuenta de AWS individual, se agrega una entrada de concesión a la ACL con el ID de usuario canónico de la cuenta.

> **Note**
>
> Si hace su bucket público (no se recomienda), cualquier usuario sin autenticar puede cargar objetos al bucket. Estos usuarios anónimos no tienen una Cuenta de AWS. Cuando un usuario anónimo carga un objeto en su bucket, Amazon S3 agrega un ID de usuario canónico especial (`65a011a29cdf8ec533ec3d1ccaae921c`) como propietario del objeto en la ACL. Para obtener más información, consulte Propiedad de los buckets y objetos de Amazon S3.

Grupos predefinidos de Amazon S3

Amazon S3 tiene un conjunto de grupos predefinidos. Al conceder acceso a la cuenta a un grupo, especifique uno de los URI de Amazon S3 en lugar de un ID de usuario canónico. Amazon S3 proporciona los siguientes grupos predefinidos:

- Grupo Usuarios autenticados: representado por `http://acs.amazonaws.com/groups/global/AuthenticatedUsers`.

Este grupo representa a todas las Cuentas de AWS. El permiso de acceso a este grupo permite que cualquier Cuenta de AWS acceda al recurso. Sin embargo, todas las solicitudes deben estar firmadas (autenticadas).

> **Warning**
>
> Cuando otorga acceso al grupo de usuarios autenticados, cualquier usuario autenticado de AWS en el mundo puede acceder a su recurso.

- Grupo Todos los usuarios: representado por `http://acs.amazonaws.com/groups/global/AllUsers`.

  El permiso de acceso a este grupo permite que cualquier persona en el mundo tenga acceso al recurso. Las solicitudes pueden estar firmadas (autenticadas) o pueden no incluir una firma (anónimas). Las solicitudes sin firmar omiten el encabezado de autenticación en la solicitud.

  > **Warning**
  >
  > Recomendamos encarecidamente que no conceda nunca los permisos `WRITE`, `WRITE_ACP` o `FULL_CONTROL` al grupo Todos los usuarios. Por ejemplo, aunque los permisos de `WRITE` no permiten que los no propietarios sobrescriban o eliminen objetos existentes, los permisos de `WRITE` siguen permitiendo a cualquier persona almacenar objetos en el bucket, que se facturan. Para obtener más información sobre estos permisos, consulte la sección siguiente ¿Qué permisos puedo conceder?.

- Grupo Envío de archivos de registro: representado por `http://acs.amazonaws.com/groups/s3/LogDelivery`.

  El permiso `WRITE` en un bucket le permite a este grupo escribir registros de acceso al servidor (consulte Registro de solicitudes con registro de acceso al servidor) en el bucket.

> **Note**
>
> Cuando utiliza ACL, el beneficiario puede ser una Cuenta de AWS o uno de los grupos predefinidos de Amazon S3. Sin embargo, el beneficiario no puede ser un usuario de IAM.

Para obtener más información acerca de los permisos y los usuarios de AWS en IAM, consulte Uso de AWS Identity and Access Management.

¿Qué permisos puedo conceder?

En la siguiente tabla se muestra el conjunto de permisos que Amazon S3 admite en una ACL. El conjunto de permisos de ACL es el mismo para una ACL de objetos y una ACL de buckets. Sin embargo, según el contexto (ACL de bucket o ACL de objeto) estos permisos de ACL conceden permisos para operaciones de buckets o de objeto específicas. La tabla muestra los permisos y describe qué significan en el contexto de objetos y buckets.

Para obtener más información acerca de los permisos de ACL en la consola de Amazon S3, consulte Configuración de la ACL.

Permisos de ACL

Permiso	Cuando se concede en un bucket	Cuando se concede en un objeto
`READ`	Le permite al beneficiario crear una lista de objetos en el bucket	Le permite al beneficiario leer los datos del objeto y sus metadatos
`WRITE`	Permite al beneficiario crear nuevos objetos en el bucket. Para los propietarios de buckets y objetos existentes, también permite eliminar y sobrescribir dichos objetos	No aplicable
`READ_ACP`	Le permite al beneficiario leer la ACL de bucket	Le permite al beneficiario leer la ACL de objeto
`WRITE_ACP`	Le permite al beneficiario escribir la ACL para el bucket correspondiente	Le permite al beneficiario escribir la ACL para el objeto correspondiente
`FULL_CONTROL`	Permite conceder los permisos `READ`, `WRITE`, `READ_ACP` y `WRITE_ACP` en el bucket	Permite conceder los permisos `READ`, `READ_ACP` y `WRITE_ACP` en el bucket

> **Warning**
>
> Extreme las precauciones a la hora de conceder permisos de acceso a sus objetos y buckets de S3. Por ejemplo, la concesión de acceso `WRITE` a un bucket permite al beneficiario crear objetos en el bucket. Se recomienda que lea toda la sección Información general de las Listas de control de acceso (ACL) antes de conceder permisos.

Mapeo de permisos de ACL y permisos de política de acceso

Como se muestra en la tabla anterior, una ACL permite solo un conjunto limitado de permisos, en comparación con el número de permisos que puede configurar en una política de acceso (consulte Acciones de políticas para Amazon S3). Cada uno de estos permisos permite una o más operaciones de Amazon S3.

La siguiente tabla muestra cómo cada permiso de ACL se asigna a los permisos de política de acceso correspondientes. Como puede ver, la política de acceso permite más permisos que ACL. ACL se utiliza principalmente para conceder permisos básicos de lectura/escritura, similar a los permisos del sistema de archivos. Para obtener más información acerca de cuándo utilizar la ACL, consulte Administración de identidades y accesos para Amazon S3.

Para obtener más información acerca de los permisos de ACL en la consola de Amazon S3, consulte Configuración de la ACL.

Permiso de ACL	Permisos de política de acceso correspondientes cuando se concede un permiso de ACL en un bucket	Permisos de política de acceso correspondientes cuando se concede un permiso de ACL en un objeto
`READ`	`s3:ListBucket` , `s3:ListBucketVersions` , y `s3:ListBucketMultipartUploads`	`s3:GetObject` y `s3:GetObjectVersion`
`WRITE`	`s3:PutObject` El propietario del bucket puede crear, sobrescribir y eliminar cualquier objeto del bucket, y el propietario del objeto tiene `FULL_CONTROL` sobre su objeto.	No aplicable

Permiso de ACL	Permisos de política de acceso correspondientes cuando se concede un permiso de ACL en un bucket	Permisos de política de acceso correspondientes cuando se concede un permiso de ACL en un objeto
	Además, cuando el beneficiario es el propietario del bucket, conceder permisos `WRITE` en una ACL de bucket permite que la acción `s3:DeleteObjectVersion` se realice en cualquier versión en ese bucket.	
`READ_ACP`	`s3:GetBucketAcl`	`s3:GetObjectAcl` y `s3:GetObjectVersionAcl`
`WRITE_ACP`	`s3:PutBucketAcl`	`s3:PutObjectAcl` y `s3:PutObjectVersionAcl`
`FULL_CONTROL`	Equivalente a otorgar permisos de ACL `READ`, `WRITE`, `READ_ACP` y `WRITE_ACP`. Por consiguiente, este permiso de ACL se asigna a una combinación de permisos de política de acceso correspondientes.	Equivalente a otorgar permisos de ACL `READ`, `READ_ACP` y `WRITE_ACP`. Por consiguiente, este permiso de ACL se asigna a una combinación de permisos de política de acceso correspondientes.

Claves de condición

Cuando concede permisos de directiva de acceso, puede utilizar claves de condición para restringir el valor de la ACL en un objeto mediante una directiva de bucket. Las siguientes claves de contexto corresponden a las ACL. Puede utilizar estas claves de contexto para ordenar el uso de una ACL específica en una solicitud:

- `s3:x-amz-grant-read`: requerir acceso de lectura.
- `s3:x-amz-grant-write`: requerir acceso de escritura.
- `s3:x-amz-grant-read-acp`: requerir acceso de lectura a la ACL del bucket.
- `s3:x-amz-grant-write-acp`: requerir acceso de escritura a la ACL del bucket.
- `s3:x-amz-grant-full-control`: requerir control total.

- `s3:x-amz-acl`: requerir un ACL predefinidas.

Por ejemplo, para las políticas que implican encabezados específicos de ACL, consulte Concesión de permiso s3:PutObject con una condición que solicita que el propietario del bucket tenga control total. Para obtener más información sobre claves de condición específicas de Amazon S3, consulte Actions, resources, and condition keys for Amazon S3 en la Referencia de autorización de servicios.

Valores de **`aclRequired`** para solicitudes comunes de Amazon S3

Para identificar las solicitudes de Amazon S3 que requerían ACL para la autorización, puede utilizar el valor `aclRequired` de los registros de acceso al servidor de Amazon S3 o AWS CloudTrail. El valor `aclRequired` que aparece en CloudTrail o en los registros de acceso al servidor de Amazon S3 depende de las operaciones a las que se haya llamado y de cierta información sobre el solicitante, el propietario del objeto y el propietario del bucket. Si no se requerían ACL, si está estableciendo la ACL predefinida `bucket-owner-full-control` o si las solicitudes están permitidas por su política de buckets, la cadena de valor de `aclRequired` es "`-`"en los registros de acceso al servidor de Amazon S3 y falta en CloudTrail.

Las siguientes tablas muestran los valores `aclRequired` esperados en los registros de acceso al servidor de CloudTrail o Amazon S3 para las distintas operaciones de la API de Amazon S3. Puede utilizar esta información para saber qué operaciones de Amazon S3 dependen de las ACL para su autorización. En las tablas siguientes, A, B y C representan las diferentes cuentas asociadas al solicitante, al propietario del objeto y al propietario del bucket. Las entradas con un asterisco (*) indican cualquiera de las cuentas A, B o C.

> **Note**
>
> Las operaciones `PutObject` de la tabla siguiente, a menos que se especifique lo contrario, indican solicitudes que no establecen una ACL, a menos que se trate de una ACL `bucket-owner-full-control`. Un valor nulo para `aclRequired` indica que `aclRequired` falta en los registros de AWS CloudTrail.

Valores de **`aclRequired`** para CloudTrail

Nombre de operación	Solicitante	Propietario del objeto	Propietario del bucket	La política de buckets concede acceso	Valor de **`aclRequired`**	Motivo
`GetObject`	A	A	A	Yes o No	null	Acceso en la misma cuenta
	A	B	A	Yes o No	null	Acceso a la misma cuenta con el propietario del bucket aplicado
	A	A	B	Sí	null	Acceso entre cuentas a través de la política de buckets
	A	A	B	No	Sí	El acceso entre cuentas se basa en la ACL
	A	A	B	Sí	null	Acceso entre cuentas a través de la política de buckets

Nombre de operación	Solicitante	Propietario del objeto	Propietario del bucket	La política de buckets concede acceso	Valor de **`aclRequired`**	Motivo
	A	B	B	No	Sí	El acceso entre cuentas se basa en la ACL
	A	B	C	Sí	null	Acceso entre cuentas a través de la política de buckets
	A	B	C	No	Sí	El acceso entre cuentas se basa en la ACL
`PutObject`	A	No aplicable	A	Yes o No	null	Acceso en la misma cuenta
	A	No aplicable	B	Sí	null	Acceso entre cuentas a través de la política de buckets

Nombre de operación	Solicitante	Propietario del objeto	Propietario del bucket	La política de buckets concede acceso	Valor de **aclRequired**	Motivo
	A	No aplicable	B	No	Sí	El acceso entre cuentas se basa en la ACL
`PutObject` con una ACL (excepto para `bucket owner-full-control`)	*	No aplicable	*	Yes o No	Sí	La solicitud concede ACL
`ListObjects`	A	No aplicable	A	Yes o No	null	Acceso en la misma cuenta
	A	No aplicable	B	Sí	null	Acceso entre cuentas a través de la política de buckets
	A	No aplicable	B	No	Sí	El acceso entre cuentas se basa en la ACL

Nombre de operación	Solicitante	Propietario del objeto	Propietario del bucket	La política de buckets concede acceso	Valor de **aclRequired**	Motivo
DeleteObject	A	No aplicable	A	Yes o No	null	Acceso en la misma cuenta
	A	No aplicable	B	Sí	null	Acceso entre cuentas a través de la política de buckets
	A	No aplicable	B	No	Sí	El acceso entre cuentas se basa en la ACL
PutObjectAcl	*	*	*	Yes o No	Sí	La solicitud concede ACL
PutBucketAcl	*	No aplicable	*	Yes o No	Sí	La solicitud concede ACL

Note

Las operaciones REST.PUT.OBJECT de la tabla siguiente, a menos que se especifique lo contrario, indican solicitudes que no establecen una ACL, a menos que se trate de una ACL bucket-owner-full-control. Una cadena de valores aclRequired de "-" indica un valor nulo en los registros de acceso al servidor de Amazon S3.

Valores **`aclRequired`** para los registros de acceso al servidor de Amazon S3

Nombre de operación	Solicitante	Propietario del objeto	Propietario del bucket	La política de buckets concede acceso	Valor de **`aclRequired`**	Motivo
`REST.GET.OBJECT`	A	A	A	Yes o No	-	Acceso en la misma cuenta
	A	B	A	Yes o No	-	Acceso a la misma cuenta con el propietario del bucket aplicado
	A	A	B	Sí	-	Acceso entre cuentas a través de la política de buckets
	A	A	B	No	Sí	El acceso entre cuentas se basa en la ACL
	A	B	B	Sí	-	Acceso entre cuentas a través de la política de buckets

Nombre de operación	Solicitante	Propietario del objeto	Propietario del bucket	La política de buckets concede acceso	Valor de **aclRequired**	Motivo
	A	B	B	No	Sí	El acceso entre cuentas se basa en la ACL
	A	B	C	Sí	-	Acceso entre cuentas a través de la política de buckets
	A	B	C	No	Sí	El acceso entre cuentas se basa en la ACL
REST.PUT.OBJECT	A	No aplicable	A	Yes o No	-	Acceso en la misma cuenta
	A	No aplicable	B	Sí	-	Acceso entre cuentas a través de la política de buckets

Nombre de operación	Solicitante	Propietario del objeto	Propietario del bucket	La política de buckets concede acceso	Valor de **aclRequired**	Motivo
	A	No aplicable	B	No	Sí	El acceso entre cuentas se basa en la ACL
REST.PUT.OBJECT con una ACL (excepto para bucket owner-full-control)	*	No aplicable	*	Yes o No	Sí	La solicitud concede ACL
REST.GET.BUCKET	A	No aplicable	A	Yes o No	-	Acceso en la misma cuenta
	A	No aplicable	B	Sí	-	Acceso entre cuentas a través de la política de buckets
	A	No aplicable	B	No	Sí	El acceso entre cuentas se basa en la ACL

Nombre de operación	Solicitante	Propietario del objeto	Propietario del bucket	La política de buckets concede acceso	Valor de **aclRequired**	Motivo
REST.DELETE.OBJECT	A	No aplicable	A	Yes o No	-	Acceso en la misma cuenta
	A	No aplicable	B	Sí	-	Acceso entre cuentas a través de la política de buckets
	A	No aplicable	B	No	Sí	El acceso entre cuentas se basa en la ACL
REST.PUT.ACL	*	*	*	Yes o No	Sí	La solicitud concede ACL

ACL de muestra

La siguiente ACL de muestra en un bucket identifica al propietario del recurso y un conjunto de concesiones. El formato es la representación XML de una ACL en la API de REST de Amazon S3. El propietario del bucket tiene FULL_CONTROL del recurso. Además, la ACL muestra cómo se otorgan los permisos para un recurso a dos Cuentas de AWS identificadas por el ID de usuario canónico, y a dos de los grupos predefinidos de Amazon S3, analizados en la sección anterior.

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>
```

ACL predefinidas

Amazon S3 admite un conjunto de concesiones predefinidas, conocidas como ACL predefinidas. Cada ACL predefinida tiene un conjunto predefinido de beneficiarios y permisos. En la siguiente tabla se muestra el conjunto de ACL predefinidas y las concesiones predefinidas asociadas.

ACL predefinidas	Se aplica a	Permisos añadidos a la ACL
`private`	Bucket y objeto	El propietario tiene `FULL_CONTROL` . Nadie más tiene derechos de acceso (opción predeterminada).
`public-read`	Bucket y objeto	El propietario tiene `FULL_CONTROL` . El grupo `AllUsers` (consulte ¿Quién es un beneficiario?) obtiene acceso `READ`.
`public-read-write`	Bucket y objeto	El propietario tiene `FULL_CONTROL` . El grupo `AllUsers` obtiene acceso `READ` y `WRITE`. Por lo general, no se recomienda conceder esto en un bucket.
`aws-exec-read`	Bucket y objeto	El propietario tiene `FULL_CONTROL` . Amazon EC2 obtiene acceso `READ` a `GET` para obtener un paquete de Amazon Machine Image (AMI) de Amazon S3.
`authenticated-read`	Bucket y objeto	El propietario tiene `FULL_CONTROL` . El grupo `AuthenticatedUsers` obtiene acceso de `READ`.
`bucket-owner-read`	Objeto	El propietario del objeto tiene `FULL_CONTROL` . El propietario del bucket obtiene acceso `READ`. Si especific a esta ACL predefinida cuando crea un bucket, Amazon S3 la ignora.
`bucket-owner-full-control`	Objeto	Tanto el propietario del objeto como el propietario del bucket tienen `FULL_CONTROL` del objeto. Si especific

ACL predefinidas	Se aplica a	Permisos añadidos a la ACL
		a esta ACL predefinida cuando crea un bucket, Amazon S3 la ignora.
`log-delivery-write`	Bucket	El grupo `LogDelivery` obtiene permisos de `WRITE` y `READ_ACP` en el bucket. Para obtener más información acerca de los logs, consulte (Registro de solicitudes con registro de acceso al servidor).

> **Note**
>
> Puede especificar solo una de estas ACL predefinidas en su solicitud.

Puede especificar una ACL enlatada en su solicitud mediante el encabezado de solicitud `x-amz-acl`. Cuando Amazon S3 recibe una solicitud con una ACL predefinida en la solicitud, añade las concesiones predefinidas a la ACL del recurso.

Configuración de la ACL

En esta sección, se explica cómo administrar permisos de acceso para buckets y objetos de S3 con listas de control de acceso (ACL). Puede agregar concesiones a la ACL de recursos con la AWS Management Console, AWS Command Line Interface (CLI), la API de REST o los SDK de AWS.

Los permisos para los buckets y los objetos son independientes entre sí. Un objeto no hereda los permisos del bucket en el que se encuentra. Por ejemplo, si crea un bucket y concede permisos de escritura a un usuario, no puede obtener acceso a los objetos de ese usuario a no ser que este le conceda acceso explícitamente.

Puede otorgar permisos a otros usuarios de Cuenta de AWS o a grupos predefinidos. El usuario o grupo al que le concede permisos se denomina beneficiario. De forma predeterminada, el propietario, que es la Cuenta de AWS que creó el bucket, tiene permisos completos.

Cada permiso que concede a un usuario o grupo añade una entrada a la ACL que está asociada con el bucket. La ACL incluye los permisos concedidos, que identifican al beneficiario y permiso concedido.

S3 Object Ownership es una configuración de bucket de Amazon S3 que puede usar para controlar la propiedad de los objetos que se cargan en el bucket y para activar o desactivar las ACL. De forma predeterminada, la propiedad de objetos se establece en la configuración impuesta por el propietario del bucket. Además, todas las ACL están deshabilitadas. Cuando las ACL están deshabilitadas, el propietario del bucket posee todos los objetos del bucket y administra su acceso de forma exclusiva mediante políticas de administración de acceso.

La mayoría de los casos de uso modernos de Amazon S3 ya no requieren el uso de ACL. Le recomendamos desactivar las ACL, excepto en circunstancias inusuales en las que necesite controlar el acceso a cada objeto de manera individual. Si las ACL están desactivadas, puede usar políticas para controlar el acceso a todos los objetos del bucket, independientemente de quién haya subido los objetos al bucket. Para obtener más información, consulte Control de la propiedad de los objetos y desactivación de las ACL del bucket.

Important

Si el bucket utiliza la configuración de propietario del bucket obligatorio de S3 Object Ownership, debe utilizar políticas para conceder acceso al bucket y a los objetos que contiene. Si la configuración impuesta por el propietario del bucket está activada, las solicitudes de configuración o actualización de las listas de control de acceso (ACL) fallan y devuelven el código de error `AccessControlListNotSupported`. Las solicitudes de lectura de ACL siguen siendo compatibles.

Warning

Le recomendamos encarecidamente que evite otorgar acceso de escritura al grupo Everyone (public access) (Todos [acceso público]) o al grupo Authenticated Users group (all AWS authenticated users) (Grupo de usuarios autenticados [todos los usuarios autenticados de AWS]). Para obtener más información sobre los efectos de conceder acceso de escritura a estos grupos, consulte Grupos predefinidos de Amazon S3.

Uso de la consola S3 para establecer permisos de ACL para un bucket

En la consola, se muestran las concesiones de acceso combinadas para los beneficiarios duplicados. Para ver la lista completa de ACL, utilice la API de REST de Amazon S3, la AWS CLI, o bien los SDK de AWS.

En la tabla siguiente se muestran los permisos de ACL que puede configurar para buckets en la consola de Amazon S3.

Permisos de ACL de la consola de Amazon S3 para buckets

Permiso de la consola	Permiso de ACL	a los datos
Objetos: lista	`READ`	Le permite al beneficiario crear una lista de objetos en el bucket
Objetos: escribir	`WRITE`	Permite al beneficiario crear nuevos objetos en el bucket. Para los propietarios de bucket y objetos de objetos existentes, también permite eliminar y sobrescribir esos objetos.
ACL del bucket: leer	`READ_ACP`	Le permite al beneficiario leer la ACL de bucket
Bucket de ACL: escribir	`WRITE_ACP`	Le permite al beneficiario escribir la ACL para el bucket correspondiente.
Todos (acceso público): objetos - lista	`READ`	Otorga acceso público de lectura para los objetos del bucket. Cuando concede acceso a la lista a Todos (acceso público), cualquier persona puede acceder a los objetos en el bucket.
Todos (acceso público): ACL del bucket - Leer	`READ_ACP`	Otorga acceso público de lectura para el ACL del bucket. Cuando concede acceso de lectura a Todos (acceso público), cualquier persona puede acceder al ACL del bucket.

Para obtener más información acerca de los permisos de ACL, consulte Información general de las Listas de control de acceso (ACL).

Important

Si el bucket utiliza la configuración de propietario del bucket obligatorio de S3 Object Ownership, debe utilizar políticas para conceder acceso al bucket y a los objetos que contiene. Si la configuración impuesta por el propietario del bucket está activada, las

solicitudes de configuración o actualización de las listas de control de acceso (ACL) fallan y devuelven el código de error `AccessControlListNotSupported`. Las solicitudes de lectura de ACL siguen siendo compatibles.

Para establecer permisos de ACL de un bucket

1. Inicie sesión en la AWS Management Console y abra la consola de Amazon S3 en https://console.aws.amazon.com/s3/.
2. En la lista Buckets (Buckets), elija el nombre del bucket para el que desea configurar permisos.
3. Elija Permissions (Permisos).
4. En Access control list (Lista de control de acceso), elija Edit (Editar).

   Puede editar los siguientes permisos de ACL para el bucket:

   Objects

   - List (Lista): permite a un beneficiario enumerar los objetos del bucket.
   - Write (Escritura): permite al beneficiario crear nuevos objetos en el Bucket. Para los propietarios de bucket y objetos de objetos existentes, también permite eliminar y sobrescribir esos objetos.

     En la consola de S3, solo puede otorgar acceso de escritura al grupo de entrega de registros de S3 y al propietario del bucket (su Cuenta de AWS). Recomendamos encarecidamente que no conceda acceso de escritura a otros beneficiarios. Sin embargo, si necesita otorgar acceso de escritura a otros beneficiarios, puede utilizar la AWS CLI, los SDK de AWS o la API de REST.

   ACL del bucket

   - Read (Lectura): permite al beneficiario leer la ACL del bucket.
   - Write (Escritura): permite al beneficiario escribir la ACL para el bucket aplicable.
5. Para cambiar los permisos del propietario del bucket, al lado de Bucket owner (your Cuenta de AWS) (Propietario del bucket [la Cuenta de AWS]), desactive o seleccione alguno de los siguientes permisos de ACL:

   - Objetos: List (Lista) o Write (Escribir)

- ACL de bucket: Read (Lectura) o Write (Escritura)

El propietario hace referencia a Usuario raíz de la cuenta de AWS, no a un usuario de IAM de AWS Identity and Access Management. Para obtener más información acerca del usuario raíz, consulte El Usuario raíz de la cuenta de AWS en la Guía del usuario de IAM.

6. Para otorgar o quitar permisos para el público en general (todos en Internet), al lado de Everyone (public access) (Todos [acceso público]), desactive o seleccione uno de los siguientes permisos de ACL:

   - Objects (Objetos): List (Lista)
   - Bucket ACL (ACL del bucket): Read (Leer)

   > ⚠ Warning
   >
   > Extreme las precauciones a la hora de otorgar al grupo Everyone (Todos) acceso público a su bucket de S3. Al otorgar acceso a este grupo, cualquier persona puede acceder a su bucket. Se recomienda encarecidamente que no otorgue nunca ningún tipo de acceso de escritura público en su bucket de S3.

7. Para otorgar o quitar permisos a cualquier persona con una Cuenta de AWS, al lado de Authenticated Users group (anyone with an Cuenta de AWS) (Grupo de usuarios autenticados [cualquiera que tenga una Cuenta de AWS]), desactive o seleccione alguno de los siguientes permisos de ACL:

   - Objects) (Objetos: List (Lista)
   - Bucket ACL) (ACL del bucket): Read (Leer)

8. Para conceder o quitar permisos para que Amazon S3 escriba registros de acceso al servidor en el bucket, en el S3 log delivery group (Grupo de entrega de registros de S3), desactive o seleccione uno de los siguientes permisos de ACL:

   - Objetos: List (Lista) o Write (Escribir)
   - ACL de bucket: Read (Lectura) o Write (Escritura)

   Si un bucket está configurado como bucket de destino para recibir registros de acceso, los permisos del bucket deben permitir al grupo Log Delivery (Envío de registros) acceso de escritura al bucket. Cuando se activa el registro de acceso al servidor en un bucket, la consola

de Amazon S3 concede acceso de escritura al grupo Log Delivery (Envío de registros) para el bucket de destino que se ha elegido para recibir los registros. Para obtener más información sobre el registro de acceso del servidor, consulte Habilitación del registro de acceso al servidor de Amazon S3.

9. Para otorgar acceso a otra Cuenta de AWS, realice lo siguiente:

   a. Elija Add grantee (Agregar beneficiario).

   b. En el cuadro Grantee (Beneficiario), ingrese el ID canónico de la otra Cuenta de AWS.

   c. Seleccione uno de los siguientes permisos de ACL:

      - Objetos: List (Lista) o Write (Escribir)
      - ACL de bucket: Read (Lectura) o Write (Escritura)

> **Warning**
>
> Cuando otorgue a otras Cuentas de AWS acceso a sus recursos, tenga en cuenta que las Cuentas de AWS pueden delegar sus permisos a los usuarios de sus cuentas. Esto se conoce como acceso entre cuentas. Para obtener información acerca del uso del acceso entre cuentas, consulte Creación de un rol para delegar permisos a un usuario de IAM en la guía del usuario de IAM.

10. Para eliminar el acceso de otra Cuenta de AWS, en Access for other Cuentas de AWS (Acceso para otras Cuentas de AWS), seleccione Remove (Eliminar).

11. Para guardar los cambios, elija Save changes (Guardar cambios).

Uso de la consola S3 para establecer permisos de ACL para un objeto

En la consola, se muestran las concesiones de acceso combinadas para los beneficiarios duplicados. Para ver la lista completa de ACL, utilice la API de REST de Amazon S3, la AWS CLI, o bien los SDK de AWS. En la tabla siguiente se muestran los permisos de ACL que puede configurar para objetos en la consola de Amazon S3.

Permisos de ACL de consola de Amazon S3 para objetos

Permiso de la consola	Permiso de ACL	a los datos
Objeto - Leer	`READ`	Le permite al beneficiario leer los datos del objeto y sus metadatos.
ACL de objeto - Leer	`READ_ACP`	Le permite al beneficiario leer la ACL de objeto.
ACL de objeto - Escribir	`WRITE_ACP`	Le permite al beneficiario escribir la ACL para el objeto correspondiente

Para obtener más información acerca de los permisos de ACL, consulte Información general de las Listas de control de acceso (ACL).

> **Important**
>
> Si el bucket utiliza la configuración de propietario del bucket obligatorio de S3 Object Ownership, debe utilizar políticas para conceder acceso al bucket y a los objetos que contiene. Si la configuración impuesta por el propietario del bucket está activada, las solicitudes de configuración o actualización de las listas de control de acceso (ACL) fallan y devuelven el código de error `AccessControlListNotSupported`. Las solicitudes de lectura de ACL siguen siendo compatibles.

Para configurar permisos de un objeto

1. Inicie sesión en la AWS Management Console y abra la consola de Amazon S3 en https://console.aws.amazon.com/s3/.
2. En la lista Buckets (Buckets), elija el nombre del bucket que contiene el objeto.
3. En la lista Objects (Objetos), seleccione el nombre del objeto para el que desea configurar los permisos.
4. Elija Permissions.
5. En lista de control de acceso (ACL), elija Edit (Editar).

Puede editar los siguientes permisos de ACL para el objeto:

Objeto

- Lectura : permite al beneficiario leer los datos del objeto y los metadatos.

ACL de objeto

- Lectura : permite al beneficiario leer la ACL del objeto.
- Escritura : permite al beneficiario escribir la ACL para el objeto aplicable. En la consola de S3, solo puede otorgar acceso de escritura al propietario del bucket (su Cuenta de AWS). Recomendamos encarecidamente que no conceda acceso de escritura a otros beneficiarios. Sin embargo, si necesita otorgar acceso de escritura a otros beneficiarios, puede utilizar la AWS CLI, los SDK de AWS o la API de REST.

6. Puede administrar permisos de acceso a objetos para lo siguiente:

   a. Acceso para el propietario del objeto

      El propietario hace referencia a Usuario raíz de la cuenta de AWS y no a un usuario de IAM de AWS Identity and Access Management. Para obtener más información acerca del usuario raíz, consulte El Usuario raíz de la cuenta de AWS en la Guía del usuario de IAM.

      Para cambiar los permisos de acceso a objetos del propietario, en Access for object owner (Acceso para el propietario del objeto), elija Your AWS Account (owner) (Su cuenta de AWS [propietario]).

      Marque las casillas de verificación para los permisos que desea cambiar y, a continuación, elija Save (Guardar).

   b. Acceso para otras Cuentas de AWS

      Para otorgar permisos a un usuario de AWS desde una Cuenta de AWS diferente, en Access for other Cuentas de AWS (Acceso para otras Cuentas de AWS), elija Add account (Agregar cuenta). En el campo Enter an ID (Ingresar un ID), ingrese el ID canónico del usuario de AWS al que desea otorgar permisos para el objeto. Para obtener información acerca de cómo buscar un ID canónico, consulte Identificadores de Cuenta de AWS en la Referencia general de Amazon Web Services. Puede añadir hasta 99 usuarios.

Tilde las casillas de verificación para los permisos que desea conceder al usuario y luego seleccione Save (Guardar). Para ver información acerca de los permisos, seleccione los iconos de ayuda.

c. Acceso público

Para conceder acceso a un objeto al público en general (a todo el mundo), en Public access (Acceso público), elija Everyone (Todos). Si se conceden permisos de acceso público, cualquier persona puede acceder al objeto desde cualquier lugar.

Marque las casillas de verificación para los permisos que desea conceder y, a continuación, elija Save (Guardar).

> **Warning**
>
> - Extreme las precauciones a la hora de otorgar al grupo Everyone (Todos) acceso anónimos a sus objetos de Amazon S3. Al otorgar acceso a este grupo, cualquier persona puede acceder a su objeto. Si necesita otorgar acceso a todo el mundo, le recomendamos encarecidamente que solo conceda permisos para Read objects (Leer objetos).
> - Recomendamos encarecidamente que no conceda permisos de objeto de escritura sobre objetos al grupo Everyone (Todos). Al hacerlo permitirá que cualquier persona sobrescriba los permisos de ACL del objeto.

Uso de los AWS SDK

Esta sección ofrece ejemplos de cómo configurar las concesiones de la lista de control de acceso (ACL) para buckets y objetos.

> **Important**
>
> Si el bucket utiliza la configuración de propietario del bucket obligatorio de S3 Object Ownership, debe utilizar políticas para conceder acceso al bucket y a los objetos que contiene. Si la configuración impuesta por el propietario del bucket está activada, las solicitudes de configuración o actualización de las listas de control de acceso (ACL) fallan

y devuelven el código de error `AccessControlListNotSupported`. Las solicitudes de lectura de ACL siguen siendo compatibles.

Java

Esta sección ofrece ejemplos de cómo configurar las concesiones de la lista de control de acceso (ACL) para buckets y objetos. El primer ejemplo crea un bucket con una ACL predefinida (consulte ACL predefinidas), crea una lista de concesiones de permisos personalizadas y luego reemplaza la ACL predefinida con una ACL que contiene las concesiones personalizadas. En el segundo ejemplo se muestra cómo modificar una ACL con el método `AccessControlList.grantPermission()`.

Example Crear un bucket y especificar una ACL predefinida que conceda permiso al grupo de entrega del registro de S3

Este ejemplo crea un bucket. En la solicitud, el ejemplo especifica una ACL predefinida que concede permiso al grupo Envío de archivos de registro para escribir registros en el bucket.

```
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 Actualizar la ACL en un objeto existente

En este ejemplo se actualiza la ACL en un objeto. En el ejemplo se realizan las siguientes tareas:

- Recuperar una ACL de un objeto
- Eliminar la ACL mediante la eliminación de todos los permisos existentes
- Añadir dos permisos: acceso completo para el propietario y WRITE_ACP (consulte ¿Qué permisos puedo conceder?) para un usuario identificado por una dirección de correo electrónico
- Guardar la ACL en el objeto

```
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 Crear un bucket y especificar una ACL predefinida que conceda permiso al grupo de entrega del registro de S3

Este ejemplo de código C# crea un bucket. En la solicitud, el código también especifica una ACL predefinida que concede permisos al grupo Envío de archivos de registro para escribir registros en el bucket.

Para obtener información acerca de cómo configurar y ejecutar ejemplos de código, consulte Introducción al SDK de AWS para .NET en la Guía para desarrolladores del SDK de AWS para .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 Actualizar la ACL en un objeto existente

En este ejemplo de código C# se actualiza la ACL en un objeto existente. En el ejemplo se realizan las siguientes tareas:

- Recuperar una ACL de un objeto.
- Eliminar la ACL mediante la eliminación de todos los permisos existentes.
- Añadir dos permisos: acceso completo para el propietario y WRITE_ACP para un usuario identificado por una dirección de correo electrónico.
- Guardar la ACL enviando una solicitud `PutAcl`.

Para obtener información acerca de cómo configurar y ejecutar ejemplos de código, consulte Introducción al SDK de AWS para .NET en la Guía para desarrolladores del SDK de AWS para .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());
            }
        }
    }
}
```

Uso de la API de REST

Las API de Amazon S3 le permiten configurar una ACL cuando crea un bucket o un objeto. Amazon S3 también proporciona una API para configurar una ACL en un bucket o un objeto existente. Estas API le proporcionan los siguientes métodos para configurar una ACL:

- Configuración de ACL con encabezados de solicitud: cuando envía una solicitud para crear un recurso (bucket u objeto), usted configura una ACL con los encabezados de solicitud. Con estos encabezados, usted puede especificar una ACL predefinida o especificar concesiones de forma explícita (identificación explícita del beneficiario y los permisos).
- Configuración de ACL con cuerpo de solicitud: cuando envía una solicitud para configurar una ACL en un recurso existente, puede configurar la ACL en el encabezado de solicitud o en el cuerpo.

Para obtener información sobre la compatibilidad con la API de REST para administrar las ACL, consulte las siguientes secciones en la referencia de la API de Amazon Simple Storage Service:

- GET Bucket acl
- PUT Bucket acl
- GET Object acl
- PUT Object acl
- PUT Object
- PUT Bucket
- PUT Object - Copy
- Initiate Multipart Upload

> **Important**
>
> Si el bucket utiliza la configuración de propietario del bucket obligatorio de S3 Object Ownership, debe utilizar políticas para conceder acceso al bucket y a los objetos que contiene. Si la configuración impuesta por el propietario del bucket está activada, las solicitudes de configuración o actualización de las listas de control de acceso (ACL) fallan y devuelven el código de error `AccessControlListNotSupported`. Las solicitudes de lectura de ACL siguen siendo compatibles.

Encabezados de solicitud específicos de la lista de control de acceso (ACL)

Puede utilizar encabezados para conceder permisos en función de listas de control de acceso (ACL). De forma predeterminada, todos los objetos son privados. Solo el propietario tiene control de acceso total. Si agrega un objeto nuevo, puede otorgar permisos a Cuentas de AWS individuales o a grupos predefinidos de Amazon S3. Estos permisos se añaden a la lista de control de acceso (ACL) del objeto. Para obtener más información, consulte Información general de las Listas de control de acceso (ACL).

Con esta operación, puede conceder permisos de acceso mediante uno de estos dos métodos:

- ACL predefinidas (**`x-amz-acl`**): Amazon S3 admite un conjunto de ACL conservadas, conocidas como "ACL predefinidas". Cada ACL predefinida tiene un conjunto predefinido de beneficiarios y permisos. Para obtener más información, consulte ACL predefinidas.

- Permisos de acceso: para otorgar permisos de acceso de forma explícita a Cuentas de AWS o grupos específicos, utilice los encabezados siguientes. Cada encabezado se asigna a permisos específicos que Amazon S3 admite en una ACL. Para obtener más información, consulte Información general de las Listas de control de acceso (ACL). En el encabezado, especifique una lista de beneficiarios que obtienen el permiso específico.
  - x-amz-grant-read
  - x-amz-grant-write
  - x-amz-grant-read-acp
  - x-amz-grant-write-acp
  - x-amz-grant-full-control

Uso de la AWS CLI

Para obtener más información acerca de la administración de las ACL mediante la AWS CLI, consulte put-bucket-acl en la Referencia de comandos de la AWS CLI.

> **Important**
>
> Si el bucket utiliza la configuración de propietario del bucket obligatorio de S3 Object Ownership, debe utilizar políticas para conceder acceso al bucket y a los objetos que contiene. Si la configuración impuesta por el propietario del bucket está activada, las solicitudes de configuración o actualización de las listas de control de acceso (ACL) fallan y devuelven el código de error `AccessControlListNotSupported`. Las solicitudes de lectura de ACL siguen siendo compatibles.

Ejemplos de políticas para ACL

Puede utilizar claves de condición en las políticas de bucket para controlar el acceso a Amazon S3.

Temas

- Concesión de permiso s3:PutObject con una condición que solicita que el propietario del bucket tenga control total
- Concesión de permiso s3:PutObject con una condición en el encabezado x-amz-acl

Concesión de permiso s3:PutObject con una condición que solicita que el propietario del bucket tenga control total

La operación PUT Object permite encabezados específicos para la lista de control de acceso (ACL) que puede utilizar para conceder permisos basados en la ACL. Con estas claves, el propietario del bucket puede configurar una condición para solicitar permisos de acceso específicos cuando el usuario carga un objeto.

Supongamos que la cuenta A tiene un bucket y el administrador de la cuenta desea conceder permisos para cargar objetos a Dave, usuario en la cuenta B. De forma predeterminada, los objetos que carga Dave pertenecen a la cuenta B y la cuenta A no tiene permisos sobre estos objetos. El propietario del bucket, que paga las facturas, desea tener los permisos completos sobre los objetos que carga Dave. Para esto, el administrador de la cuenta A puede conceder el permiso `s3:PutObject` a Dave, con la condición de que la solicitud incluya los encabezados específicos de ACL, que concede permisos totales de forma explícita o utiliza una ACL predefinida. Para obtener más información, consulte PUT Object.

Requerir el encabezado x-amz-full-control

Puede requerir el encabezado `x-amz-full-control` en la solicitud con permiso de control total para el propietario del bucket. La siguiente política de bucket concede el permiso `s3:PutObject` al usuario Dave con una condición que utiliza la clave de condición `s3:x-amz-grant-full-control`, que requiere que la solicitud incluya el encabezado `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**
>
> En este ejemplo se describe el permiso entre cuentas. No obstante, si Dave (quien recibe el permiso) pertenece a la Cuenta de AWS que posee el bucket, este permiso condicional no es necesario. Esto se debe a que la cuenta principal a la que pertenece Dave es propietaria de los objetos que carga el usuario.

Agregar denegación de forma explícita

La política de bucket anterior concede el permiso condicional al usuario Dave en la cuenta B. Mientras esta política tenga vigencia, es posible que Dave obtenga el mismo permiso sin ninguna condición mediante otra política. Por ejemplo, Dave puede pertenecer a un grupo y usted concede el permiso `s3:PutObject` al grupo sin ninguna condición. Para evitar esas ambigüedades en los permisos, puede escribir una política de acceso más estricta y añadir una denegación explícita. En este ejemplo, deniega el permiso de carga de forma explícita al usuario Dave si no incluye los encabezados necesarios en la solicitud que concede los permisos completos al propietario del bucket. La denegación explícita siempre sustituye a cualquier otro permiso concedido. A continuación se muestra el ejemplo de política de acceso revisado con denegación explícita añadida.

```
{
    "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"
                }
            }
        }
    ]
}
```

Prueba de la política con la AWS CLI

Si tiene dos Cuentas de AWS, puede probar la política mediante AWS Command Line Interface (AWS CLI). Puede asociar la política y, con las credenciales de Dave, usar el siguiente comando `put-object` de la AWS CLI para probar el permiso. Para proporcionar las credenciales de Dave, debe añadir el parámetro `--profile`. Para conceder el permiso de control total al propietario del bucket, debe añadir el parámetro `--grant-full-control`. Para obtener más información acerca de la configuración y el uso de la AWS CLI, consulte Desarrollo con Amazon S3 mediante la AWS CLI.

```
aws s3api put-object --bucket examplebucket --key HappyFace.jpg --body c:\HappyFace.jpg
 --grant-full-control id="AccountA-CanonicalUserID" --profile AccountBUserProfile
```

Requerir el encabezado x-amz-acl

Puede solicitar que el encabezado `x-amz-acl` con una ACL predefinida conceda permiso de control total al propietario del bucket. Para pedir el encabezado `x-amz-acl` en la solicitud, puede reemplazar el par clave-valor en el bloque `Condition` y especificar la clave de condición `s3:x-amz-acl`, tal y como se muestra en el siguiente ejemplo.

```
"Condition": {
    "StringEquals": {
        "s3:x-amz-acl": "bucket-owner-full-control"
    }
```

```
}
```

Para probar el permiso con la AWS CLI, debe especificar el parámetro `--acl`. Luego, la AWS CLI añade el encabezado `x-amz-acl` cuando envía la solicitud.

```
aws s3api put-object --bucket examplebucket --key HappyFace.jpg --body c:\HappyFace.jpg
 --acl "bucket-owner-full-control" --profile AccountBadmin
```

Concesión de permiso s3:PutObject con una condición en el encabezado x-amz-acl

La siguiente política de bucket concede el permiso `s3:PutObject` para dos Cuentas de AWS si la solicitud incluye el encabezado `x-amz-acl`, que permite la lectura pública del objeto. El bloque `Condition` utiliza la condición `StringEquals` y proporciona un par de clave-valor, `"s3:x-amz-acl":["public-read"]`, para evaluación. En el par clave-valor, `s3:x-amz-acl` es una clave específica de Amazon S3, como lo indica el prefijo `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**
>
> No todas las condiciones son adecuadas para todas las acciones. Por ejemplo, es adecuado incluir una condición `s3:LocationConstraint` en una política que concede el permiso `s3:CreateBucket` de Amazon S3. Sin embargo, no tiene sentido incluir esta condición en una política que concede el permiso `s3:GetObject`. Amazon S3 puede realizar pruebas de errores semánticos de este tipo que implican condiciones específicas de Amazon S3. Sin embargo, si crea una política para un usuario o rol de IAM e incluye una condición de Amazon S3 no válida semánticamente, no se reportará ningún error porque IAM no puede validar condiciones de Amazon S3.

Bloquear el acceso público a su almacenamiento de Amazon S3

La característica Block Public Access de Amazon S3 proporciona la configuración de los puntos de acceso, los buckets y las cuentas, con el fin de ayudarle a administrar el acceso público a los recursos de Amazon S3. De forma predeterminada, los buckets, puntos de acceso y objetos nuevos no permiten el acceso público. Sin embargo, los usuarios pueden modificar las políticas de bucket, las políticas de punto de acceso o los permisos de objeto para permitir el acceso público. La configuración de S3 Block Public Access anula estas políticas y permisos para poder limitar el acceso público a estos recursos.

Con Block Public Access de S3, los administradores de cuentas y los propietarios de buckets pueden configurar fácilmente controles centralizados para limitar el acceso público a sus recursos de Amazon S3, que se aplican independientemente de cómo se creen los recursos.

Para obtener instrucciones sobre cómo configurar el acceso público en bloque, consulte Configuración de Block Public Access.

Cuando Amazon S3 recibe una solicitud para acceder a un bucket o a un objeto, determina si el bucket o la cuenta del propietario del bucket tiene aplicada una configuración de bloqueo de acceso público. Si la solicitud se realizó a través de un punto de acceso, Amazon S3 también comprueba la configuración de bloqueo de acceso público del punto de acceso. Si hay una configuración de bloqueo de acceso público que prohíbe el acceso solicitado, Amazon S3 rechaza la solicitud.

Block Public Access de Amazon S3 proporciona cuatro configuraciones. Estas configuraciones son independientes y se pueden usar en cualquier combinación. Cada configuración se puede aplicar a un punto de acceso, a un bucket o a una Cuenta de AWS completa. Si la configuración de bloqueo

de acceso público para el punto de acceso, el bucket o la cuenta es diferente, Amazon S3 aplicará la combinación más restrictiva de la configuración del punto de acceso, el bucket y la cuenta.

Cuando Amazon S3 evalúa si una configuración de bloqueo de acceso público prohíbe una operación, rechaza cualquier solicitud que infrinja un punto de acceso, un bucket o una configuración de cuenta.

Important

El acceso público se otorga a buckets y objetos a través de listas de control de acceso (ACL), políticas de punto de acceso, políticas de bucket o todas ellas. Para ayudar a garantizar que todos los puntos de acceso, buckets y objetos de Amazon S3 tienen su acceso público bloqueado, se recomienda activar los cuatro ajustes de bloqueo de acceso público de la cuenta. Estos ajustes bloquean el acceso público a todos los buckets y puntos de acceso actuales y futuros.

Antes de aplicar estos ajustes, verifique que sus aplicaciones funcionen correctamente sin acceso público. Si necesita algún nivel de acceso público a los buckets u objetos, como, por ejemplo, con el fin de alojar un sitio web estático, tal y como se describe en Alojamiento de un sitio web estático mediante Amazon S3, puede personalizar los ajustes individuales para que se adapten a sus casos de uso de almacenamiento.

Habilitar Bloqueo de acceso público ayuda a proteger sus recursos al impedir que el acceso público se conceda a través de las políticas de recursos o las listas de control de acceso (ACL) que se adjuntan directamente a los recursos de S3. Además de habilitar Bloqueo de acceso público, examine detenidamente las siguientes políticas para confirmar que no conceden acceso público:

- Políticas basadas en identidades adjuntas a las entidades principales de AWS asociadas (por ejemplo, los roles de IAM)
- Políticas basadas en recursos adjuntas a recursos de AWS asociados (por ejemplo, claves de AWS Key Management Service (KMS))

Note

- Puede habilitar la configuración de bloqueo de acceso público solo para los puntos de acceso, los buckets y las Cuentas de AWS. Amazon S3 no admite la configuración de bloqueo de acceso público por objeto.

- Cuando aplica la configuración de bloqueo de acceso público a una cuenta, esta se aplica globalmente a todas las Regiones de AWS. La configuración podría no entrar en vigor en todas las regiones de manera inmediata o simultánea, pero acabará propagándose a todas las regiones.

Temas

Configurar Block Public Access

S3 Block Public Access proporciona cuatro configuraciones. Puede aplicar esta configuración en cualquier combinación en puntos de acceso o buckets individuales o en Cuentas de AWS completas. Si aplica una configuración a una cuenta, se aplica a todos los buckets y puntos de acceso propiedad de esa cuenta. Del mismo modo, si aplica una configuración a un bucket, se aplica a todos los puntos de acceso asociados a ese bucket.

La siguiente tabla contiene las configuraciones disponibles

Nombre	Descripción
`BlockPublicAcls`	Si se establece esta opción en `TRUE` causa el siguiente comportamiento: • Las llamadas de la acl de PUT Object y la acl de PUT Object fallan si la lista de control de acceso (ACL) es pública. • Las llamadas de objetos de PUT fallan si la solicitud incluye una ACL pública.

Nombre	Descripción
	• Si se aplica esta configuración a una cuenta, las llamadas al PUT Bucket fallan si la solicitud incluye una ACL pública. Cuando esta configuración se establece en TRUE, se produce un error en las operaciones especificadas (independientemente de si se hacen a través de la API de REST, la AWS CLI o los SDK de AWS). No obstante, las políticas existentes y las ACL para buckets y objetos no se modifican . Esta configuración permite protegerse frente al acceso público al tiempo que le permite auditar, ajustar o alterar de otro modo las políticas existentes y ACL de sus buckets y objetos. **Note** Los puntos de acceso no tienen ACL asociadas a ellos. Si aplica esta configuración a un punto de acceso, actúa como paso a través del bucket subyacente. Si un punto de acceso tiene esta configura ción habilitada, las solicitudes realizadas a través del punto de acceso se comportan como si el bucket subyacente tuviera esta configuración habilitada, independientemente de si el bucket la tiene habilitada o no.

Nombre	Descripción
`IgnorePublicAcls`	Al configurar esta opción en `TRUE`, Amazon S3 ignora todas las ACL públicas en un bucket y cualquier objeto que contenga. Esta configuración le permite bloquear de manera segura el acceso público concedido por las ACL al tiempo que permite llamadas a PUT Object que incluyen una ACL pública (frente a la `BlockPublicAcls` que rechaza llamadas a PUT Object que incluyen una ACL pública). La habilitación de esta configuración no afecta a la persistencia de ninguna ACL existente y no evita el establecimiento de ACL públicas nuevas. **Note** Los puntos de acceso no tienen ACL asociadas a ellos. Si aplica esta configuración a un punto de acceso, actúa como paso a través del bucket subyacente. Si un punto de acceso tiene esta configuración habilitada, las solicitudes realizadas a través del punto de acceso se comportan como si el bucket subyacente tuviera esta configuración habilitada, independientemente de si el bucket la tiene habilitada o no.

Nombre	Descripción
`BlockPublicPolicy`	La configuración de esta opción en `TRUE` para un bucket hace que Amazon S3 rechace llamadas a la política de buckets de PUT si la política de buckets especificada permite el acceso público. Establecer esta opción en `TRUE` para un bucket también provoca que Amazon S3 rechace llamadas a la política de punto de acceso PUT para todos los puntos de acceso entre cuentas del bucket si la política especificada permite el acceso público. Si se establece esta opción en `TRUE` para un punto de acceso, Amazon S3 rechazará las llamadas a la política de punto de acceso PUT y a la política de bucket PUT que se realicen a través del punto de acceso si la política especificada (ya sea para el punto de acceso o para el bucket subyacente) permite el acceso público. Puede usar esta configuración para permitir a los usuarios administrar políticas de punto de acceso y de bucket sin permitirles compartir públicamente el bucket ni los objetos que contiene. Habilitar esta configuración no afecta a las políticas de punto de acceso o bucket existentes. **Important** Para usar esta configuración de manera eficaz, le recomendamos que la aplique en el nivel de cuenta. Una política de bucket puede permitir a los usuarios alterar la configuración de Block Public Access de un bucket. Por tanto, los usuarios que tienen permiso para cambiar una política de bucket podrían insertar una política que les permita deshabilitar la configuración de Block Public Access del bucket. Si se habilita esta configuración para toda la cuenta en lugar de para un bucket específico, entonces Amazon S3 bloquea políticas públicas incluso si un usuario altera la política del bucket para deshabilitar esta configuración.

Nombre	Descripción
`RestrictPublicBuckets`	El establecimiento de esta opción en `TRUE` restringe el acceso a un punto de acceso o un bucket con una política pública a solo las entidades principales del servicio de AWS y los usuarios autorizados dentro de la cuenta del propietario del bucket y la cuenta del propietario del punto de acceso. Esta configuración bloquea todo el acceso entre cuentas al punto de acceso o al bucket (salvo a las entidades principales de servicios de AWS), pero sigue permitiendo que los usuarios de la cuenta administren el punto de acceso o el bucket. La activación de este ajuste no afecta a las políticas de punto de acceso o bucket existentes, salvo que Amazon S3 bloquea el acceso público y entre cuentas derivado de cualquier política de punto de acceso o bucket pública, incluida la delegación no pública a cuentas específicas.

Important

- Las llamadas a GET Bucket acl y GET Object acl siempre devuelve los permisos efectivos instaurados para el bucket u objeto especificados. Por ejemplo, imagine que un bucket tiene un ACL que concede acceso público, pero el bucket también tiene la configuración `IgnorePublicAcls` habilitada. En este caso, GET Bucket acl devuelve una ACL que refleja los permisos de acceso que Amazon S3 está implementando, en lugar de la ACL real asociada con el bucket.
- La configuración de Block Public Access no altera las políticas ni las ACL existentes. Por tanto, la eliminación de una configuración de Block Public Access provoca que pueda accederse de nuevo públicamente a un bucket o un objeto con una política pública o ACL.

Realización de operaciones de acceso público de bloque en un punto de acceso

Para realizar operaciones de Block Public Access en un punto de acceso, use el servicio de la AWS CLI `s3control`.

> **Important**
>
> Tenga en cuenta que actualmente no es posible cambiar la configuración de bloqueo de acceso público de un punto de acceso después de haberlo creado. Por lo tanto, la única forma de especificar la configuración de bloqueo de acceso público para un punto de acceso es incluirla al crear el punto de acceso.

Qué significa "pública"

ACL

Amazon S3 considera que una ACL de objetos o buckets es pública si concede cualquier permiso a los miembros de los grupos `AllUsers` o `AuthenticatedUsers` definidos previamente. Para obtener más información acerca de los grupos predefinidos, consulte Grupos predefinidos de Amazon S3.

Políticas de buckets

Al evaluar una política de bucket, Amazon S3 comienza suponiendo que la política es pública. A continuación, evalúa la política para determinar si califica como no pública. Para que se considere no pública, una política de bucket debe conceder acceso solo a valores fijos (valores que no contienen un comodín o una política variable de AWS Identity and Access Management) para uno o más de los siguientes:

- Una entidad principal de AWS, un usuario, un rol o una entidad principal de servicio (por ejemplo, `aws:PrincipalOrgID`)
- Un conjunto de Classless Inter-Domain Routings (CIDR), mediante `aws:SourceIp`. Para obtener más información sobre CIDR, consulte RFC 4632 en la página web de RFC EditoR.

> **Note**
>
> Las políticas de buckets que otorgan acceso condicionado a la clave de condición `aws:SourceIp` con rangos de IP muy amplios (por ejemplo, 0.0.0.0/1) se evalúan como “públicas”. Esto incluye valores superiores a /8 para IPv4 y /32 para IPv6 (excluidos los rangos privados de RFC1918). Bloquear el acceso público rechazará estas políticas “públicas” e impedirá el acceso entre cuentas a los grupos que ya utilizan estas políticas “públicas”.

- `aws:SourceArn`
- `aws:SourceVpc`
- `aws:SourceVpce`
- `aws:SourceOwner`
- `aws:SourceAccount`
- `s3:x-amz-server-side-encryption-aws-kms-key-id`
- `aws:userid`, fuera del patrón "`AROLEID:*`"
- `s3:DataAccessPointArn`

> **Note**
>
> Cuando se utiliza en una política de bucket, este valor puede contener un comodín para el nombre del punto de acceso sin mostrar la política pública, siempre y cuando el ID de cuenta sea fijo. Por ejemplo, permitir el acceso a `arn:aws:s3:us-west-2:123456789012:accesspoint/*` permitiría el acceso a cualquier punto de acceso asociado con la cuenta `123456789012` de la región `us-west-2` sin hacer pública la política de bucket. Tenga en cuenta que este comportamiento es diferente para las políticas de punto de acceso. Para obtener más información, consulte Puntos de acceso.

- `s3:DataAccessPointAccount`

Para obtener más información acerca de las políticas de bucket, consulte Políticas de buckets para Amazon S3.

Example : políticas de buckets públicas

En virtud de estas reglas, las siguientes políticas de ejemplo se consideran públicas:

```
{
  "Principal": "*",
  "Resource": "*",
  "Action": "s3:PutObject",
  "Effect": "Allow"
 }
```

```
{
  "Principal": "*",
```

```
  "Resource": "*",
  "Action": "s3:PutObject",
  "Effect": "Allow",
  "Condition": { "StringLike": {"aws:SourceVpc": "vpc-*"}}
 }
```

Puede convertir estas políticas en no públicas incluyendo cualquiera de las claves de condición indicadas con anterioridad, utilizando un valor fijo. Por ejemplo, puede convertir la última política de arriba en no pública estableciendo `aws:SourceVpc` en un valor fijo, como el siguiente:

```
{
  "Principal": "*",
  "Resource": "*",
  "Action": "s3:PutObject",
  "Effect": "Allow",
  "Condition": {"StringEquals": {"aws:SourceVpc": "vpc-91237329"}}
 }
```

Cómo evalúa Amazon S3 una política de bucket que contiene concesiones de acceso público y no público

Este ejemplo muestra cómo evalúa Amazon S3 una política de bucket que contiene concesiones de acceso público y no público.

Imagine que un bucket tiene una política que concede acceso a un conjunto de entidades principales fijas. Bajo las reglas descritas con anterioridad, esta política no es pública. Por tanto, si habilita la configuración `RestrictPublicBuckets`, la política permanece en efecto tal y como está redactada, porque `RestrictPublicBuckets` solo se aplica a buckets que tienen políticas públicas. Sin embargo, si añade una instrucción pública a la política, `RestrictPublicBuckets` surtirá efecto en el bucket. Solo permite que las entidades principales del servicio de AWS y los usuarios autorizados de la cuenta del propietario del bucket tengan acceso al bucket.

Suponga, por ejemplo, que un bucket propiedad de "Cuenta-1 tiene una política que contiene lo siguiente:

1. Una instrucción que concede acceso a AWS CloudTrail (que es una entidad principal de servicio de AWS)
2. Una instrucción que concede acceso a la cuenta "Cuenta-2"
3. Una instrucción que concede acceso al público, especificando, por ejemplo, `"Principal": "*"` sin `Condition` limitante

Esta política califica como pública debido a la tercera instrucción. Con esta política en vigor y `RestrictPublicBuckets` activado, Amazon S3 solo permite obtener acceso a CloudTrail. Aunque la instrucción 2 no es pública, Amazon S3 deshabilita el acceso de la "Cuenta-2". Esto se debe a que la instrucción 3 convierte en pública a toda la política, por lo que se aplica `RestrictPublicBuckets`. Como consecuencia, Amazon S3 deshabilita el acceso entre cuentas, aunque la política delega el acceso a una cuenta específica, "Cuenta-2". Pero si elimina la instrucción 3 de la política, la política no califica como pública y `RestrictPublicBuckets` ya no es aplicable. Así, "Cuenta-2" recupera el acceso al bucket, aunque deje `RestrictPublicBuckets` habilitado.

Puntos de acceso

Amazon S3 evalúa la configuración del bloqueo de acceso público de un modo ligeramente diferente para los puntos de acceso en comparación con los buckets. Las reglas que Amazon S3 aplica para determinar cuándo una política de punto de acceso es pública suelen ser las mismas para los puntos de acceso que para los buckets, excepto en las situaciones siguientes:

- Un punto de acceso que tiene un origen de red VPC siempre se considera no público, independientemente del contenido de su política de punto de acceso.
- Una política de punto de acceso que concede acceso a un conjunto de puntos de acceso que utilizan `s3:DataAccessPointArn` se considera pública. Tenga en cuenta que este comportamiento es diferente al de las políticas de bucket. Por ejemplo, una política de bucket que concede acceso a los valores de `s3:DataAccessPointArn` que coinciden con `arn:aws:s3:us-west-2:123456789012:accesspoint/*` no se considera pública. Sin embargo, la misma instrucción en una política de punto de acceso haría público el punto de acceso.

Uso de Analizador de acceso de IAM para S3 para revisar los buckets públicos

Puede utilizar Analizador de acceso de IAM para S3 para revisar los buckets con ACL de bucket, las políticas de bucket o las políticas de punto de acceso que otorgan acceso público. Analizador de acceso de IAM para S3 le avisa de los buckets que están configurados para permitir el acceso a cualquier usuario de Internet u otras Cuentas de AWS, incluidas aquellas Cuentas de AWS ajenas a la organización. Para cada bucket público o compartido, recibirá resultados que le informarán del origen y el nivel de acceso público o compartido.

En Analizador de acceso de IAM para S3, puede bloquear todo el acceso público a un bucket con un solo clic. También puede examinar a fondo las configuraciones de permisos de bucket para

configurar niveles pormenorizados de acceso. Para casos de uso específicos y verificados que requieren acceso público o compartido, puede reconocer y registrar su intención de que el bucket continúe siendo público o compartido archivando los resultados del bucket.

En casos excepcionales, Analizador de acceso de IAM para S3 podría no tener ningún resultado para un bucket que una evaluación del bloqueo de acceso público de Amazon S3 registre como pública. Esto sucede porque el bloqueo de acceso público de Amazon S3 revisa las políticas de las acciones actuales y todas las acciones posibles que podrían añadirse en el futuro, lo que hace que un bucket se convierta en público. Por otro lado, Analizador de acceso de IAM para S3 solo analiza las acciones actuales especificadas para el servicio de Amazon S3 en la evaluación del estado de acceso.

Para obtener más información acerca de Analizador de acceso de IAM para S3, consulte Revisión del acceso al bucket mediante Analizador de acceso de IAM para S3.

Permisos

Para usar características del bloqueo de acceso público de Amazon S3, debe contar con los siguientes permisos:

Operation	Permisos necesarios
Estado de la política de bucket de GET	`s3:GetBucketPolicyStatus`
Configuración de Block Public Access de GET bucket	`s3:GetBucketPublicAccessBlock`
Configuración de Block Public Access de PUT bucket	`s3:PutBucketPublicAccessBlock`
Configuración de Block Public Access de DELETE bucket	`s3:PutBucketPublicAccessBlock`
Configuración de Block Public Access de GET account	`s3:GetAccountPublicAccessBlock`
Configuración de Block Public Access de PUT account	`s3:PutAccountPublicAccessBlock`

Operation	Permisos necesarios
Configuración de Block Public Access de DELETE account	`s3:PutAccountPublicAccessBlock`
Configuración de bloqueo de acceso público de punto de acceso PUT	`s3:CreateAccessPoint`

> **Note**
>
> Las operaciones DELETE necesitan los mismos permisos que las operaciones PUT. NO hay permisos separados para las operaciones DELETE.

Configuración de Block Public Access

Para obtener más información sobre cómo configurar el bloqueo de acceso público para su Cuenta de AWS y sus buckets de Amazon S3, consulte los siguientes temas.

- Establecer la configuración de Block Public Access para la cuenta
- Establecer la configuración de Block Public Access para sus buckets de S3

Establecer la configuración de Block Public Access para la cuenta

Block Public Access de Amazon S3 proporciona la configuración de los puntos de acceso, los buckets y las cuentas, a fin de ayudarlo a administrar el acceso público a los recursos de Amazon S3. De forma predeterminada, los buckets, puntos de acceso y objetos nuevos no permiten el acceso público.

Para obtener más información, consulte Bloquear el acceso público a su almacenamiento de Amazon S3.

> **Note**
>
> La configuración del nivel de cuenta anula la configuración de los objetos individuales. Si configura su cuenta para bloquear el acceso público, se anulará cualquier configuración de acceso público aplicada a los objetos individuales de su cuenta.

Puede utilizar la consola de S3, la AWS CLI, los SDK de AWSy la API de REST con el fin de configurar el bloqueo del acceso público para todos los buckets de la cuenta. Para obtener más información, consulte las secciones siguientes.

Para establecer la configuración de bloqueo de acceso público para sus buckets, consulte Establecer la configuración de Block Public Access para sus buckets de S3. Para obtener información acerca de los puntos de acceso, consulte Realización de operaciones de acceso público de bloque en un punto de acceso.

Uso de la consola de S3

El bloqueo del acceso público de Amazon S3 impide que se aplique cualquier ajuste que permita el acceso público a los datos dentro de los buckets de S3. En esta sección, se describe cómo editar la configuración de bloqueo del acceso público para todos los buckets de S3 en su Cuenta de AWS. Para obtener más información acerca del bloqueo del acceso público, consulte Bloquear el acceso público a su almacenamiento de Amazon S3.

Para editar la configuración de bloqueo del acceso público para todos los buckets de S3 en una Cuenta de AWS

1. Inicie sesión AWS Management Console Management Console y abra la consola de Amazon S3 en https://console.aws.amazon.com/s3/.
2. Elija Block Public Access settings for this account (Configuración de la cuenta para bloquear el acceso público).
3. Elija Edit (Editar) a fin de cambiar la configuración de bloqueo del acceso público para todos los buckets de su Cuenta de AWS.
4. Elija la configuración que desea cambiar y, a continuación, elija Save changes (Guardar cambios).
5. Cuando se le pida que confirme, introduzca **`confirm`**. Para guardar los cambios, elija Save (Guardar).

Uso de la AWS CLI

Puede utilizar el bloqueo del acceso público de Amazon S3 a través de la AWS CLI. Para obtener más información acerca de cómo configurar y usar la AWS CLI, consulte ¿Qué es la AWS Command Line Interface?

Cuenta

Para realizar operaciones de Block Public Access en una cuenta, use el servicio de la AWS CLI `s3control`. Estas son las operaciones de cuentas que utilizan este servicio:

- PUT PublicAccessBlock (para una cuenta)
- GET PublicAccessBlock (para una cuenta)
- DELETE PublicAccessBlock (para una cuenta)

Para obtener más información y ejemplos, consulte put-public-access-block en la Referencia de la AWS CLI.

Uso de los AWS SDK

Java

En los siguientes ejemplos, se muestra cómo utilizar el bloqueo del acceso público de Amazon S3 con AWS SDK for Java a fin de establecer una configuración de bloqueo de acceso público en una cuenta de 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

Este ejemplo pertenece exclusivamente a operaciones de cuenta que utilizan la clase de cliente `AWSS3Control`. Para operaciones de bucket, consulte el ejemplo anterior.

Other SDKs

Para obtener más información sobre el uso de otros AWS SDK, consulte Desarrollo con Amazon S3 mediante los SDK de AWS.

Uso de la API de REST

Para obtener información sobre el uso de Block Public Access (Bloqueo de acceso público) de Amazon S3 con las API de REST, consulte los siguientes temas en la referencia de la API de Amazon Simple Storage Service.

- Operaciones de cuenta
  - PUT PublicAccessBlock
  - GET PublicAccessBlock
  - DELETE PublicAccessBlock

Establecer la configuración de Block Public Access para sus buckets de S3

Block Public Access de Amazon S3 proporciona la configuración de los puntos de acceso, los buckets y las cuentas, a fin de ayudarlo a administrar el acceso público a los recursos de Amazon S3. De forma predeterminada, los buckets, puntos de acceso y objetos nuevos no permiten el acceso público.

Para obtener más información, consulte Bloquear el acceso público a su almacenamiento de Amazon S3.

Puede utilizar la consola de S3, AWS CLI, los SDK de AWS y la API de REST para garantizar el acceso público a uno o más buckets. También existe la posibilidad de bloquear el acceso público a los buckets que ya son públicos. Para obtener más información, consulte las secciones siguientes.

Para establecer la configuración de bloqueo de acceso público para cada bucket de la cuenta, consulte Establecer la configuración de Block Public Access para la cuenta. Para obtener información acerca de cómo configurar el acceso público de bloques para puntos de acceso, consulte Realización de operaciones de acceso público de bloque en un punto de acceso.

Uso de la consola de S3

El bloqueo del acceso público de Amazon S3 impide que se aplique cualquier ajuste que permita el acceso público a los datos dentro de los buckets de S3. Esta sección describe cómo editar

la configuración de bloqueo del acceso público para uno o más buckets de S3. Para obtener información sobre cómo bloquear el acceso público mediante la AWS CLI, los SDK de AWS y las API de REST de Amazon S3, consulte Bloquear el acceso público a su almacenamiento de Amazon S3.

Puede ver si su bucket es accesible públicamente en la lista de buckets . En la columna Access (Acceso), Amazon S3 etiqueta los permisos de un bucket de la siguiente manera:

- Public (Público): todos los usuarios tienen acceso a una o más de las siguientes opciones: listar objetos, escribir objetos y leer y escribir permisos.
- Objects can be public (Los objetos pueden ser públicos): el bucket no es público, pero todos los usuarios con los permisos pertinentes pueden otorgar acceso público a objetos.
- Buckets and objects not public (Los buckets y los objetos no son públicos): los buckets y los objetos no son de acceso público.
- Solo los usuarios autorizados de esta cuenta: el acceso se limita a los usuarios y los roles de IAM en esta cuenta y las entidades principales de servicio de AWS debido a que hay una política que otorga acceso público.

También puede filtrar la búsqueda de buckets por tipo de acceso. Elija un tipo de acceso de la lista desplegable al lado de la barra Search for buckets (Buscar buckets).

Si ve un `Error` cuando muestre los buckets y la configuración de acceso público, es posible que no tenga los permisos necesarios. Consulte para asegurarse de que dispone de los siguientes permisos agregados a la política del usuario o rol:

```
s3:GetAccountPublicAccessBlock
s3:GetBucketPublicAccessBlock
s3:GetBucketPolicyStatus
s3:GetBucketLocation
s3:GetBucketAcl
s3:ListAccessPoints
s3:ListAllMyBuckets
```

En algunos casos excepcionales, las solicitudes también pueden producir un error debido a una interrupción de Región de AWS.

Para editar la configuración de Block Public Access de Amazon S3 para un único bucket de S3

Siga estos pasos si tiene que cambiar la configuración de acceso público para un solo bucket de S3.

1. Inicie sesión en la AWS Management Console y abra la consola de Amazon S3 en https://console.aws.amazon.com/s3/.
2. En la lista Bucket name (Nombre del bucket), seleccione el nombre del bucket que desee.
3. Elija Permissions.
4. Elija Edit (Editar) para cambiar la configuración de acceso público del bucket. Para obtener más información acerca de las cuatro configuraciones de Block Public Access de Amazon S3, consulte Configurar Block Public Access.
5. Elija la configuración que desea cambiar y, a continuación, elija Save (Guardar).
6. Cuando se le pida que confirme, introduzca **`confirm`**. Para guardar los cambios, elija Save (Guardar).

Puede cambiar la configuración de bloqueo del acceso público de Amazon S3 cuando se crea un bucket. Para obtener más información, consulte Crear un bucket.

Mediante AWS CLI

Para bloquear el acceso público en un bucket o eliminar el bloque de acceso público, utilice el servicio de la AWS CLI `s3api`. Estas son las operaciones de bucket que utilizan este servicio:

- PUT PublicAccessBlock (para un bucket)
- GET PublicAccessBlock (para un bucket)
- DELETE PublicAccessBlock (para un bucket)
- GET BucketPolicyStatus

Para obtener más información y ejemplos, consulte put-public-access-block en la Referencia de la AWS CLI.

Uso de los 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**
>
> Este ejemplo pertenece exclusivamente a operaciones de bucket que utilizan la clase de cliente `AmazonS3`. Para operaciones de cuenta, consulte el siguiente ejemplo.

Other SDKs

Para obtener más información sobre el uso de otros AWS SDK, consulte Desarrollo con Amazon S3 mediante los SDK de AWS.

Uso de la API de REST

Para obtener información sobre el uso de Block Public Access (Bloqueo de acceso público) de Amazon S3 con las API de REST, consulte los siguientes temas en la referencia de la API de Amazon Simple Storage Service.

- Operaciones de bucket
  - PUT PublicAccessBlock
  - GET PublicAccessBlock
  - DELETE PublicAccessBlock
  - GET BucketPolicyStatus

Revisión del acceso al bucket mediante Analizador de acceso de IAM para S3

Analizador de acceso de IAM para S3 le avisa de los buckets de S3 que están configurados para permitir el acceso a cualquier usuario de Internet u otras Cuentas de AWS, incluidas aquellas Cuentas de AWS ajenas a la organización. Para cada bucket público o compartido, recibe los resultados sobre el origen y el nivel del acceso público o compartido. Por ejemplo, Analizador

de acceso de IAM para S3 puede mostrar que un bucket tiene acceso de lectura o escritura proporcionado a través de una lista de control de acceso (ACL) de bucket, una política de bucket o una política de punto de acceso. Con estos resultados, puede adoptar medidas correctivas inmediatas y precisas para restaurar el acceso al bucket a aquel que se pretende.

Al revisar un bucket en riesgo en Analizador de acceso de IAM para S3, puede bloquear todo el acceso público al bucket con un solo clic. Le recomendamos que bloquee todo el acceso a sus buckets a menos que necesite acceso público para admitir un caso de uso específico. Antes de bloquear todo el acceso público, asegúrese de que las aplicaciones seguirán funcionando correctamente sin ese acceso público. Para obtener más información, consulte Bloquear el acceso público a su almacenamiento de Amazon S3.

También puede examinar a fondo las configuraciones de permisos de bucket para configurar niveles pormenorizados de acceso. Para casos de uso específicos y verificados que requieren acceso público, como el alojamiento estático de sitios web, descargas públicas o uso compartido entre cuentas, puede reconocer y registrar su intención de que el bucket siga siendo público o compartido archivando los resultados del bucket. Puede volver a visitar y modificar estas configuraciones de bucket en cualquier momento. También puede descargar sus resultados en un informe CSV con fines de auditoría.

Analizador de acceso de IAM para S3 está disponible sin coste adicional en la consola de Amazon S3. Analizador de acceso de IAM para S3 cuenta con la tecnología de Analizador de acceso de AWS Identity and Access Management (IAM). Para utilizar Analizador de acceso de IAM para S3 en la consola de Amazon S3, debe ir a la consola de IAM y habilitar Analizador de acceso de IAM por región.

Para obtener más información acerca de Analizador de acceso de IAM, consulte Analizador de acceso de IAM en la Guía del usuario de IAM. Para obtener más información acerca de Analizador de acceso de IAM para S3, revise las secciones siguientes.

Important

- para S3 requiere un analizador en el nivel de cuenta. Para utilizar Analizador de acceso de IAM para S3, debe ir a Analizador de acceso de IAM y crear un analizador que tenga una cuenta como zona de confianza. Para obtener más información, consulte Habilitación de Analizador de acceso de IAM en la guía del usuario de IAM.
- Analizador de acceso de IAM para S3 no analiza la política de puntos de acceso que se adjunta a los puntos de acceso entre cuentas. Este comportamiento se produce porque el

punto de acceso y su política están fuera de la zona de confianza, es decir, de la cuenta. Los buckets que delegan el acceso a un punto de acceso entre cuentas se muestran en Buckets with public access (Buckets con acceso público) si no ha aplicado la configuración de bloqueo del acceso público `RestrictPublicBuckets` al bucket o a la cuenta. Al aplicar la configuración de bloqueo del acceso público `RestrictPublicBuckets`, el bucket aparece en Buckets con acceso desde otras Cuentas de AWS — incluidas Cuentas de AWS de terceros.

- Cuando se agrega o modifica una política de bucket o una ACL de bucket, Analizador de acceso de IAM genera y actualiza los resultados basándose en el cambio en un plazo de 30 minutos. Es posible que los resultados relacionados con la configuración de acceso público del nivel de cuenta no se generen ni actualicen hasta 6 horas después de haber cambiado la configuración. Es posible que los resultados relacionados con los puntos de acceso multirregionales no se generen ni actualicen hasta seis horas después de que se cree, elimine o cambie la política del punto de acceso multirregional.

Temas

¿Qué información proporciona Analizador de acceso de IAM para S3?

Analizador de acceso de IAM para S3 proporciona información sobre los buckets a los que se puede acceder fuera de su Cuenta de AWS. Cualquier usuario de Internet puede acceder a los buckets enumerados en Buckets with public access (Buckets con acceso público). Si Analizador de acceso de IAM para S3 identifica buckets públicos, también muestra una advertencia en la parte superior de la página, con el número de buckets públicos de la región. Los buckets enumerados en Buckets with access from other Cuentas de AWS — including third-party Cuentas de AWS (Buckets con acceso

desde otras Cuentas de AWS, incluidas las Cuentas de AWS de terceros) se comparten de manera condicionada con otras Cuentas de AWS, incluidas las cuentas ajenas a su organización.

Para cada bucket, Analizador de acceso de IAM para S3 proporciona la siguiente información:

- Nombre del bucket
- Detectado por Analizador de acceso: cuando Analizador de acceso de IAM para S3 detecta el acceso público o compartido a los buckets.
- Compartido a través de: indica cómo se comparte el bucket, a través de una política de bucket, de una ACL de bucket o de ambas. Los puntos de acceso multirregionales y los puntos de acceso entre cuentas se reflejan en los puntos de acceso. Un bucket se puede compartir a través de políticas y ACL. Si desea buscar y revisar el origen del acceso al bucket, puede utilizar la información de esta columna como punto de partida para adoptar medidas correctivas inmediatas y precisas.
- Estado: el estado del resultado del bucket. Analizador de acceso de IAM para S3 muestra los resultados de todos los buckets públicos y compartidos.
  - Activo: no se ha revisado el resultado.
  - Archivado: el resultado se ha revisado y confirmado como previsto.
  - Todos: todos los resultados para los buckets públicos o compartidos con otras Cuentas de AWS, incluidas las Cuentas de AWS ajenas a la organización.
- Nivel de acceso: permisos de acceso concedidos para el bucket:
  - Lista: permite enumerar recursos.
  - Lectura: permite leer pero no editar el contenido y los atributos de los recursos.
  - Escritura: permite crear, eliminar o modificar recursos.
  - Permisos: permite conceder o modificar permisos de recursos.
  - Etiquetado: permite actualizar las etiquetas asociadas con el recurso.

Habilitación de Analizador de acceso de IAM para S3

Para usar Analizador de acceso de IAM para S3 debe completar los siguientes requisitos previos.

1. Concesión de permisos necesarios.

   Para obtener más información, consulte Permisos necesarios para usar Analizador de acceso de IAM en la guía del usuario de IAM.

2. Vaya a IAM para crear un analizador de cuenta para cada región en la que desee utilizar Analizador de acceso de IAM.

   Analizador de acceso de IAM para S3 requiere un analizador en el nivel de cuenta. Para utilizar Analizador de acceso de IAM para S3, debe crear un analizador que tenga una cuenta como zona de confianza. Para obtener más información, consulte Habilitación de Analizador de acceso de IAM en la guía del usuario de IAM.

Bloquear todo el acceso público

Si desea bloquear todo el acceso a un bucket con un solo clic, puede utilizar el botón Bloquear todo el acceso público en Analizador de acceso de IAM para S3. Cuando se bloquea todo el acceso público a un bucket, no se concede ningún acceso público. Recomendamos bloquear todo el acceso público a los buckets, a menos que se necesite acceso público para admitir un caso de uso específico y comprobado. Antes de bloquear todo el acceso público, asegúrese de que las aplicaciones seguirán funcionando correctamente sin ese acceso público.

Si no desea bloquear todo el acceso público al bucket, puede editar la configuración del bloqueo de acceso público en la consola de Amazon S3 para configurar niveles pormenorizados de acceso a los buckets. Para obtener más información, consulte Bloquear el acceso público a su almacenamiento de Amazon S3.

En casos excepcionales, Analizador de acceso de IAM para S3 podría no tener ningún resultado para un bucket que una evaluación del bloqueo de acceso público de Amazon S3 registre como pública. Esto sucede porque el bloqueo de acceso público de Amazon S3 revisa las políticas de las acciones actuales y todas las acciones posibles que podrían añadirse en el futuro, lo que hace que un bucket se convierta en público. Por otro lado, Analizador de acceso de IAM para S3 solo analiza las acciones actuales especificadas para el servicio de Amazon S3 en la evaluación del estado de acceso.

Para bloquear todo el acceso público a un bucket mediante Analizador de acceso de IAM para S3

1. Inicie sesión AWS Management Console Management Console y abra la consola de Amazon S3 en https://console.aws.amazon.com/s3/.
2. En el panel de navegación de la izquierda, en Dashboards (Paneles), elija Access analyzer for S3 (Analizador de acceso para S3).
3. En Analizador de acceso de IAM para S3, elija un bucket.
4. Elija Block all public access (Bloquear todo el acceso público).

5. Para confirmar su intención de bloquear todo el acceso público al bucket, en Block all public access (bucket settings) (Bloquear todo el acceso público (configuración del bucket)), escriba **`confirm`**.

   Amazon S3 bloquea todo el acceso público a su bucket. El estado de los resultados del bucket se actualiza a resuelto y el bucket desaparece del listado de Analizador de acceso de IAM para S3. Si desea revisar los buckets resueltos, abra Analizador de acceso de IAM en la consola de IAM.

Revisar y cambiar el acceso al bucket

Si no tiene intención de conceder acceso a las cuentas públicas u otras cuentas de Cuentas de AWS, incluidas las que son ajenas a la organización, puede modificar la ACL del bucket, la política del bucket o la política del punto de acceso para eliminar el acceso al bucket. La columna Shared through (Compartido a través de) muestra todos los orígenes de acceso a bucket: política de bucket, ACL de bucket y/o política de punto de acceso. Los puntos de acceso multirregionales y los puntos de acceso entre cuentas se reflejan en los puntos de acceso.

Para revisar y cambiar una política de bucket, una ACL de bucket o una política de punto de acceso

1. Abra la consola de Amazon S3 en https://console.aws.amazon.com/s3.
2. En el panel de navegación, elija Access analyzer for S3 (Analizador de acceso para S3).
3. Para ver si se ha concedido acceso público o compartido mediante una política de bucket, una ACL de bucket o una política de punto de acceso, busque en la columna Shared through (Compartido a través de).
4. En Buckets, elija el nombre del bucket con la política de bucket, la ACL del bucket o la política de punto de acceso que desee cambiar o revisar.
5. Si desea cambiar o ver una ACL de bucket:

   a. Elija Permissions (Permisos).

   b. Elija Access Control List (Lista de control de acceso [ACL]).

   c. Revise la ACL del bucket y realice los cambios necesarios.

      Para obtener más información, consulte Configuración de la ACL.
6. Si desea cambiar o revisar una política de bucket:

   a. Elija Permissions (Permisos).

b. Elija Bucket Policy (Política del bucket).

c. Revise o cambie la política de bucket según sea necesario.

   Para obtener más información, consulte Agregar una política de bucket mediante la consola de Amazon S3.

7. Si desea revisar o cambiar una política de punto de acceso:

   a. Seleccionar Multi-Region Access Point (Punto de acceso para multirregiones).

   b. Elija el nombre del punto de acceso de multirregiones.

   c. Revise o cambie la política de punto de acceso para multirregiones según sea necesario.

      Para obtener más información, consulte Permisos.

8. Si desea revisar o cambiar una política de punto de acceso:

   a. Elija Access points (Puntos de acceso).

   b. Seleccione el nombre del punto de acceso.

   c. Revise o cambie el acceso según sea necesario.

      Para obtener más información, consulte Uso de puntos de acceso de Amazon S3 con la consola de Amazon S3.

   Si edita o elimina una ACL de bucket, una política de bucket o un punto de acceso para eliminar el acceso público o compartido, se actualiza el estado de los resultados del bucket, es decir, quedan resueltos. Los resultados del bucket resueltos desaparecen del listado de Analizador de acceso de IAM para S3, pero se pueden ver en Analizador de acceso de IAM.

Archivar resultados del bucket

Si un bucket concede acceso al público o a otras Cuentas de AWS, incluidas las que son ajenas a la organización, para admitir un caso de uso específico (por ejemplo: un sitio web estático, descargas públicas o uso compartido entre cuentas), puede archivar los resultados del bucket. Al archivar los resultados del bucket, usted confirma y registra su intención de que el bucket siga siendo público o compartido. Los resultados del bucket archivados permanecen en su listado de Analizador de acceso de IAM para S3 para que pueda saber en todo momento qué buckets son públicos o compartidos.

Para archivar los resultados de buckets en Analizador de acceso de IAM para S3

1. Abra la consola de Amazon S3 en https://console.aws.amazon.com/s3.
2. En el panel de navegación, elija Access analyzer for S3 (Analizador de acceso para S3).
3. En Analizador de acceso de IAM para S3, elija un bucket activo.
4. Para confirmar su intención de que el público u otras Cuentas de AWS puedan obtener acceso a este bucket, incluidas las cuentas ajenas a la organización, elija Archive (Archivar).
5. Escriba **`confirm`** y elija Archive (Archivar).

Activar los resultados de los buckets archivados

Después de archivar los resultados, en cualquier momento puede volver a consultarlos y cambiar su estado para activarlos e indicar que el bucket requiere otra revisión.

Para activar un resultado de un bucket archivado en Analizador de acceso de IAM para S3

1. Abra la consola de Amazon S3 en https://console.aws.amazon.com/s3.
2. En el panel de navegación, elija Access analyzer for S3 (Analizador de acceso para S3).
3. Elija los resultados del bucket archivado.
4. Seleccione Mark as active (Marcar como activos).

Consultar los detalles de los resultados

Si necesita más información sobre un bucket, puede abrir los detalles de los resultados del bucket en Analizador de acceso de IAM en la consola de IAM.

Para ver los detalles de los resultados en Analizador de acceso de IAM para S3

1. Abra la consola de Amazon S3 en https://console.aws.amazon.com/s3.
2. En el panel de navegación, elija Access analyzer for S3 (Analizador de acceso para S3).
3. En Analizador de acceso de IAM para S3, elija un bucket.
4. Elija View details (Ver detalles).

   Los detalles del resultado se abren en Analizador de acceso de IAM en la consola de IAM.

Descarga de un informe de Analizador de acceso de IAM para S3

Puede descargar los resultados del bucket en un informe CSV que se puede utilizar con fines de auditoría. El informe incluye la misma información que se ve en Analizador de acceso de IAM para S3 en la consola de Amazon S3.

Para descargar un informe

1. Abra la consola de Amazon S3 en https://console.aws.amazon.com/s3.
2. En el panel de navegación de la izquierda, elija Access a Analyzer for S3 (Analizador de acceso para S3).
3. En el filtro Region (Región), elija la región.

   Analizador de acceso de IAM para S3 se actualiza para mostrar los buckets de la región seleccionada.
4. Elija Download report (Descargar informe).

   Se genera un informe en formato CSV y se guarda en el equipo.

Verificación de la propiedad del bucket con la condición de propietario del bucket

La condición de propietario del bucket de Amazon S3 garantiza que los buckets que utiliza en las operaciones de S3 pertenezcan a las Cuentas de AWS que espera.

La mayoría de las operaciones de S3 se leen o escriben en buckets específicos de S3. Estas operaciones incluyen cargar, copiar y descargar objetos, recuperar o modificar configuraciones del bucket y recuperar o modificar configuraciones de objetos. Al realizar estas operaciones, especifique el bucket que desee utilizar incluyendo su nombre con la solicitud. Por ejemplo, para recuperar un objeto de S3, realice una solicitud que especifique el nombre de un bucket y la clave de objeto a partir de la que recuperar ese bucket.

Puesto que Amazon S3 identifica los buckets en función de sus nombres, una aplicación que utilice un nombre de bucket incorrecto en una solicitud podría realizar operaciones involuntarias en un bucket diferente al esperado. Para ayudar a evitar interacciones involuntarias del bucket en situaciones como esta, puede usar la condición de propietario del bucket. La condición de propietario del bucket le permite verificar que el bucket de destino sea propiedad de la Cuenta de

AWS esperada, lo que proporciona una capa adicional de seguridad de que sus operaciones de S3 tienen los efectos que desea.

Temas

Cuándo utilizar la condición de propietario del bucket

Recomendamos usar la condición de propietario del bucket siempre que realice una operación de S3 compatible y conozca el ID de cuenta del propietario del bucket esperado. La condición de propietario del bucket está disponible para todas las operaciones de objetos de S3 y la mayoría de las operaciones del bucket de S3. Para obtener una lista de las operaciones de S3 que no admiten la condición de propietario del bucket, consulte Restricciones y limitaciones.

Para ver los beneficios de usar la condición de propietario del bucket, considere el siguiente escenario en el que se involucra a la clienta de AWS Bea:

1. Bea desarrolla una aplicación que utiliza Amazon S3. Durante el desarrollo, Bea utiliza su Cuenta de AWS destinada a pruebas solamente para crear un bucket denominado `bea-data-test` y configura su aplicación de manera que realice solicitudes a `bea-data-test`.
2. Bea implementa su aplicación, pero olvida volver a configurar la aplicación para utilizar un bucket en su Cuenta de AWS de producción.
3. En producción, la aplicación de Bea realiza solicitudes en `bea-data-test`, que se producen correctamente. Esto da como resultado que los datos de producción se escriban en el bucket en la cuenta de prueba de Bea.

Bea puede servir de protección frente a situaciones como esta mediante el uso de la condición de propietario del bucket. Con la condición de propietario del bucket, Bea puede incluir el ID de la Cuenta de AWS del propietario del bucket esperado en sus solicitudes. A continuación, Amazon S3 comprueba el ID de cuenta de propietario del bucket antes de procesar cada solicitud. Si el propietario real del bucket no coincide con el propietario del bucket esperado, se produce un error en la solicitud.

Si Bea utiliza la condición de propietario del bucket, el escenario descrito anteriormente no dará como resultado que la aplicación de Bea escriba involuntariamente en un bucket de prueba. En su lugar, se producirá un error de `Access Denied` en las solicitudes que realice su aplicación en el paso 3. Gracias al uso de la condición de propietario del bucket, Bea ayuda a eliminar el riesgo de interactuar de forma accidental con los buckets en una Cuenta de AWS incorrecta.

Verificación del propietario del bucket

Para utilizar la condición de propietario del bucket, incluye un parámetro con la solicitud que especifica el propietario del bucket esperado. La mayoría de las operaciones de S3 implican un solo bucket, y solo requieren este parámetro único para usar la condición de propietario del bucket. Para las operaciones de `CopyObject`, este primer parámetro especifica el propietario esperado del bucket de destino e incluye un segundo parámetro para especificar el propietario esperado del bucket de origen.

Cuando realice una solicitud que incluya un parámetro de condición de propietario del bucket, S3 comprobará el ID de cuenta del propietario del bucket con el parámetro especificado antes de procesar la solicitud. Si el parámetro coincide con el ID de cuenta del propietario del bucket, S3 procesará la solicitud. Si el parámetro no coincide con el ID de cuenta del propietario del bucket, se producirá un error de `Access Denied` en la solicitud.

Puede utilizar la condición de propietario del bucket con AWS Command Line Interface (AWS CLI), los SDK de AWS y las API de REST de Amazon S3. Cuando utilice la condición de propietario del bucket con la AWS CLI y las API de REST de Amazon S3, utilice los siguientes nombres de parámetros.

Método de acceso	Parámetro para operaciones que no son de copia	Parámetro de origen para operaciones de copia	Parámetro de destino para operaciones de copia
AWS CLI	`--expected-bucket-owner`	`--expected-source-bucket-owner`	`--expected-bucket-owner`
API de REST de Amazon S3	Encabezado `x-amz-expected-bucket-owner`	Encabezado `x-amz-source-expected-bucket-owner`	Encabezado `x-amz-expected-bucket-owner`

Los nombres de parámetros necesarios para utilizar la condición de propietario del bucket con los AWS SDK varían según el idioma. Para determinar los parámetros requeridos, consulte la documentación del SDK para el idioma deseado. Puede encontrar la documentación del SDK en Herramientas para crear en AWS.

Ejemplos

En los siguientes ejemplos, se muestra cómo puede implementar la condición de propietario del bucket en Amazon S3 mediante la AWS CLI o AWS SDK for Java 2.x.

Example

Ejemplo: Carga de un objeto

En el siguiente ejemplo, se carga un objeto en el bucket de S3 *amzn-s3-demo-bucket1* con la condición de propietario del bucket para garantizar que *amzn-s3-demo-bucket1* sea propiedad de la Cuenta de AWS 111122223333.

AWS CLI

```
aws s3api put-object \
                --bucket amzn-s3-demo-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("amzn-s3-demo-bucket1")
            .key("exampleobject")
            .expectedBucketOwner("111122223333")
            .build();
    Path path = Paths.get("example_file.txt");
    s3Client.putObject(request, path);
}
```

Example

Ejemplo: Copia de un objeto

En el ejemplo siguiente se copia el objeto `object1` del bucket de S3 *amzn-s3-demo-bucket1* al bucket de S3 *amzn-s3-demo-bucket2*. Utiliza la condición de propietario del bucket para garantizar que los buckets son propiedad de las cuentas esperadas de acuerdo con la tabla siguiente.

Bucket	Propietario esperado
amzn-s3-demo-bucket1	111122223333
amzn-s3-demo-bucket2	444455556666

AWS CLI

```
aws s3api copy-object --copy-source amzn-s3-demo-bucket1/object1 \
                            --bucket amzn-s3-demo-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("amzn-s3-demo-bucket1/object1")
                .destinationBucket("amzn-s3-demo-bucket2")
                .destinationKey("object1copy")
                .expectedSourceBucketOwner("111122223333")
                .expectedBucketOwner("444455556666")
                .build();
        s3Client.copyObject(request);
    }
```

Example

Ejemplo: Recuperación de una política del bucket

En el siguiente ejemplo, se recupera la política de acceso para el bucket de S3 *amzn-s3-demo-bucket1* con la condición de propietario del bucket para garantizar que *amzn-s3-demo-bucket1* sea propiedad de la Cuenta de AWS 111122223333.

AWS CLI

```
aws s3api get-bucket-policy --bucket amzn-s3-demo-bucket1 --expected-bucket-
owner 111122223333
```

AWS SDK for Java 2.x

```
public void getBucketPolicyExample() {
    S3Client s3Client = S3Client.create();
    GetBucketPolicyRequest request = GetBucketPolicyRequest.builder()
            .bucket("amzn-s3-demo-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();
    }
}
```

Restricciones y limitaciones

La condición de propietario del bucket de Amazon S3 tiene las siguientes restricciones y limitaciones:

- El valor del parámetro de condición de propietario del bucket debe ser el ID de una Cuenta de AWS (valor numérico de 12 dígitos). No se admiten entidades de servicio.
- La condición de propietario del bucket no está disponible para CreateBucket, ListBuckets, ni ninguna de las operaciones incluidas en AWS S3 Control. Amazon S3 ignora los parámetros de condición de propietario del bucket incluidos en las solicitudes a estas operaciones.
- La condición de propietario del bucket solo comprueba que la cuenta especificada en el parámetro de verificación sea propietaria del bucket. La condición de propietario del bucket no comprueba la configuración del bucket. Tampoco garantiza que la configuración del bucket cumpla condiciones específicas o coincida con cualquier estado pasado.

Control de la propiedad de los objetos y desactivación de las ACL del bucket

S3 Object Ownership es una configuración de nivel de bucket de Amazon S3 que puede usar para controlar la propiedad de los objetos que se cargan en el bucket y para activar o desactivar las listas de control de acceso (ACL). De forma predeterminada, la propiedad de objetos se establece en la configuración impuesta por el propietario del bucket. Además, todas las ACL están deshabilitadas. Cuando las ACL están deshabilitadas, el propietario del bucket posee todos los objetos del bucket y administra el acceso a los datos de forma exclusiva mediante políticas de administración de acceso.

La mayoría de los casos de uso modernos de Amazon S3 ya no requieren el uso de ACL. Le recomendamos mantener las ACL desactivadas, excepto en circunstancias inusuales en las que necesite controlar el acceso de cada objeto de manera individual. Si las ACL están desactivadas, puede usar políticas para controlar el acceso más fácilmente a cada objeto del bucket, independientemente de quién haya subido los objetos al bucket.

La propiedad de objetos tiene tres configuraciones que puede utilizar para controlar la propiedad de los objetos que se cargan en el bucket y desactivar o habilitar las ACL:

ACL desactivadas

- Propietario del bucket obligatorio (predeterminado): las ACL están desactivadas y el propietario del bucket tiene automáticamente la propiedad y el control total sobre cada objeto del bucket. Las ACL ya no afectan a los permisos de los datos del bucket de S3. El bucket utiliza políticas para definir el control de acceso.

ACL habilitadas

- Propietario del bucket preferido: el propietario del bucket tiene la propiedad y el control total sobre los nuevos objetos que otras cuentas escriben en el bucket con la ACL predefinida `bucket-owner-full-control`.
- Escritor del objeto: la Cuenta de AWS que carga un objeto es propietaria del objeto, tiene control total sobre él y puede conceder a otros usuarios acceso a él a través de ACL.

Para la mayoría de los casos de uso modernos de S3, le recomendamos que desactive las ACL eligiendo la configuración de propietario del bucket obligatorio y utilice su política de bucket para compartir datos con usuarios fuera de la cuenta según sea necesario. Este enfoque simplifica la

administración de permisos. Puede desactivar las ACL tanto en los buckets recién creados como en los que ya existen. Para los buckets recién creados, las ACL están deshabilitadas de forma predeterminada. En el caso de un bucket existente que ya tiene objetos, después de desactivar las ACL, las ACL de objeto y bucket ya no forman parte de una evaluación de acceso y el acceso se concede o deniega sobre la base de políticas. Para los buckets existentes, puede volver a habilitar las ACL en cualquier momento después de desactivarlas, y las ACL de bucket y objeto preexistentes se restauran.

Antes de desactivar las ACL, le recomendamos que revise la política de bucket para asegurarse de que cubre todas las formas en que pretende conceder acceso al bucket fuera de la cuenta. Después de desactivar las ACL, el bucket solo acepta solicitudes PUT que no especifican solicitudes ACL o PUT con ACL de control total del propietario del bucket, tales como la ACL predefinida `bucket-owner-full-control` o formas equivalentes de esta ACL expresadas en XML. Las aplicaciones existentes que admiten ACL de control total del propietario del bucket no se ven afectadas. Las solicitudes PUT que contienen otras ACL (por ejemplo, concesiones personalizadas a determinadas Cuentas de AWS) producen un error y devuelve un error `400` con el código de error `AccessControlListNotSupported`.

En cambio, un bucket con la configuración de propietario del bucket preferido sigue aceptando y respetando las ACL de bucket y objeto. Con esta configuración, nuevos objetos que se escriben con la ACL predefinida `bucket-owner-full-control` pertenecen automáticamente al propietario del bucket en lugar del escritor del objeto. Todos los demás comportamientos de ACL siguen vigentes. Para requerir que todas las operaciones PUT de Amazon S3 incluyan la ACL predefinida `bucket-owner-full-control`, puede agregar una política de bucket que permita solo cargas de objetos mediante esta ACL.

Para ver qué configuración de propiedad de objetos se aplica a los buckets, puede utilizar las métricas de la Lente de almacenamiento de Amazon S3. Lente de almacenamiento de S3 es una función de análisis de almacenamiento en la nube que puede utilizar para obtener visibilidad en toda la organización sobre el uso y la actividad del almacenamiento de objetos. Para obtener más información, consulte Uso de Lente de almacenamiento de S3 para encontrar la configuración de propiedad de objetos.

> **Note**
>
> Para obtener más información sobre el uso de la clase de almacenamiento Amazon S3 Express One Zone con buckets de directorio, consulte ¿Qué es S3 Express One Zone? y Buckets de directorio.

Configuración de la propiedad de objetos

En esta tabla se muestra el impacto que cada configuración de propiedad de objetos tiene en las ACL, los objetos, la propiedad de objetos y las cargas de objetos.

Opción	Aplica a	Efecto en la propiedad de objetos	Efecto en las ACL	Cargas aceptadas
Aplicada al propietario del bucket (predeterminado)	Todos los objetos existentes y nuevos	El propietario del bucket es dueño de todos los objetos.	Las ACL están desactivadas y ya no afectan a los permisos de acceso al bucket. Las solicitudes de configuración o actualización de ACL fallan. Sin embargo, las solicitudes de lectura de ACL son compatibles. El propietario del bucket tiene plena propiedad y control. El escritor de objetos ya no tiene plena propiedad y control.	Cargas con ACL de control total del propietario del bucket o cargas que no especifican una ACL
Propietario del bucket preferido	Objetos nuevos	Si la carga de un objeto incluye la ACL predefini	Las ACL se pueden actualizar y pueden	Todas las cargas

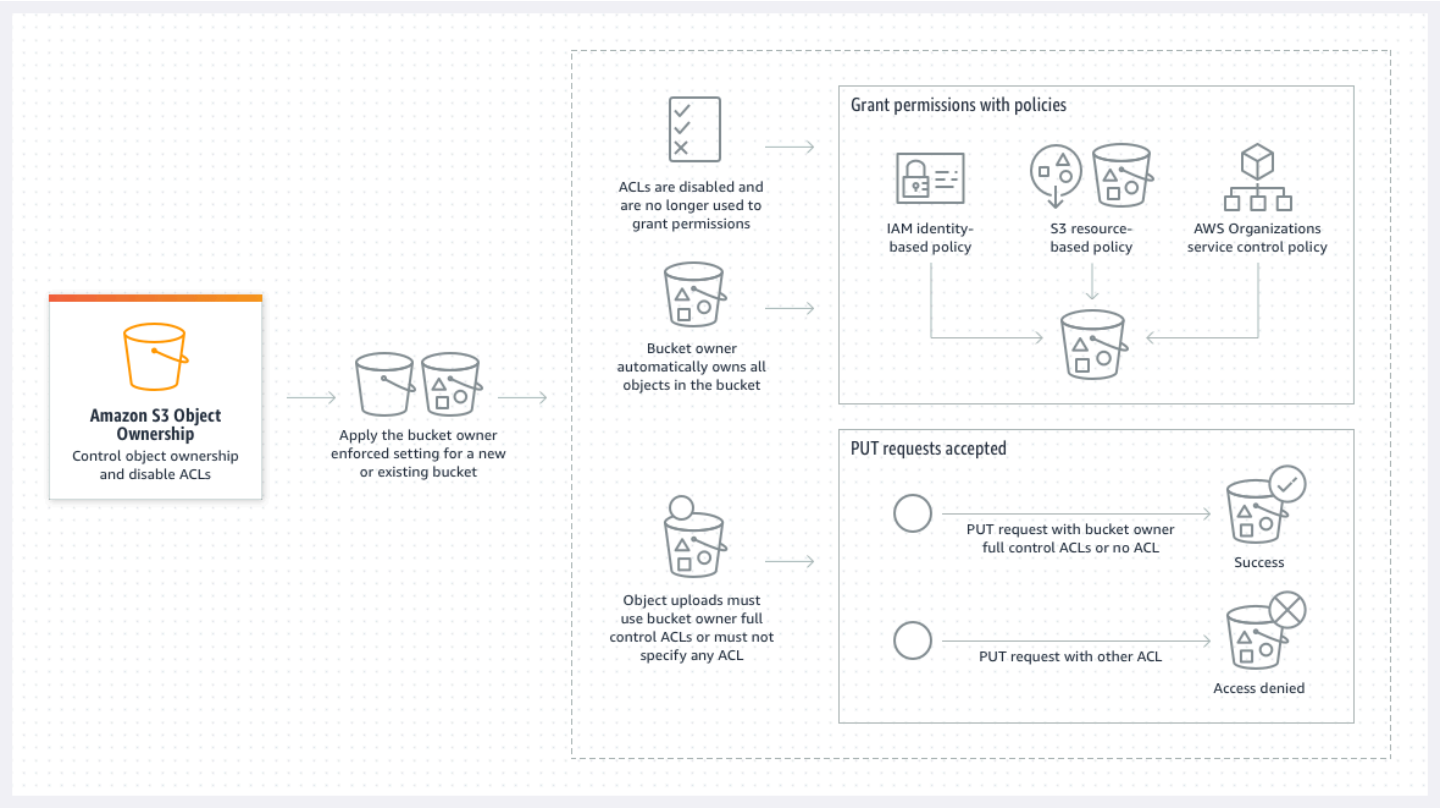
Opción	Aplica a	Efecto en la propiedad de objetos	Efecto en las ACL	Cargas aceptadas
		da `bucket-owner-full-control`, el propietario del bucket es el propietario del objeto. Los objetos cargados con otras ACL son propiedad de la cuenta de escritura.	conceder permisos. Si la carga de un objeto incluye la ACL predefinida `bucket-owner-full-control`, el propietario del bucket tiene acceso de control total y el escritor del objeto ya no tiene acceso de control total.	
Escritor de objetos	Objetos nuevos	El escritor del objeto es propietario del objeto.	Las ACL se pueden actualizar y pueden conceder permisos. El escritor del objeto tiene acceso de control total.	Todas las cargas

Cambios introducidos al desactivar las ACL

Cuando se aplica la configuración Aplicada al propietario del bucket de Propiedad de objetos para desactivar las ACL, automáticamente tiene la propiedad y el control total de cada objeto del

bucket sin realizar ninguna acción adicional. Aplicada al propietario del bucket es la configuración predeterminada para todos los buckets recién creados. Después de aplicar la configuración Aplicada al propietario del bucket, verá tres cambios:

- Todas las ACL de bucket y de objetos están desactivadas, lo que le da acceso completo como propietario del bucket. Cuando realice una solicitud de ACL de lectura en el bucket u objeto, verá que el acceso completo solo se concede al propietario del bucket.
- Como propietario del bucket, tiene automáticamente la propiedad y el control total sobre cada objeto del bucket.
- Las ACL ya no afectan a los permisos de acceso al bucket. Como resultado, el control de acceso de los datos se basa en políticas, tales como políticas de IAM, políticas de bucket de S3, políticas de puntos de conexión de VPC y políticas de control de servicios (SCP) de las organizaciones.



Si utiliza S3 Versioning, el propietario del bucket tiene la propiedad y el control total sobre todas las versiones de objetos del bucket. La aplicación de la configuración Aplicada al propietario del bucket no agrega una nueva versión de un objeto.

Los objetos nuevos se pueden cargar en el bucket solo si utilizan ACL de control total del propietario del bucket o no especifican una ACL. Las cargas de objetos fallan si especifican cualquier otra ACL. Para obtener más información, consulte Resolución de problemas.

Como la siguiente operación `PutObject` de ejemplo que utiliza la AWS Command Line Interface (AWS CLI) incluye la ACL predefinida `bucket-owner-full-control`, el objeto se puede cargar en un bucket con ACL desactivadas.

```
aws s3api put-object --bucket amzn-s3-demo-bucket --key key-name --body path-to-file --
acl bucket-owner-full-control
```

Dado que la siguiente operación `PutObject` no especifica una ACL, también se realiza correctamente para un bucket con ACL desactivadas.

```
aws s3api put-object --bucket amzn-s3-demo-bucket --key key-name --body path-to-file
```

> **Note**
>
> Si otras Cuentas de AWS necesitan acceso a objetos después de la carga, debe conceder permisos adicionales a esas cuentas mediante políticas de bucket. Para obtener más información, consulte Explicaciones que utilizan políticas para administrar el acceso a los recursos de Amazon S3.

Rehabilitación de ACL

Puede volver a habilitar las ACL cambiando la configuración Aplicada al propietario del bucket por otra configuración de propiedad de objetos en cualquier momento. Si utilizó ACL de objeto para la administración de permisos antes de aplicar la configuración Aplicada al propietario del bucket y no migró estos permisos de ACL de objetos a la política de bucket, estos permisos se restauran después de volver a habilitar las ACL. Además, los objetos escritos en el bucket mientras se aplicó la configuración Aplicada al propietario del bucket siguen siendo propiedad del propietario del bucket.

Por ejemplo, si cambia de la configuración Aplicada al propietario del bucket a Escritor de objetos, ya no tendrá automáticamente la propiedad ni el control total sobre los objetos que anteriormente eran propiedad de otras Cuentas de AWS. En su lugar, las cuentas de carga vuelven a ser propietarias de estos objetos. Los objetos que pertenecen a otras cuentas utilizan ACL para obtener permisos, por lo que no puede utilizar políticas para conceder permisos a estos objetos. Sin embargo, como

propietario del bucket, sigue siendo propietario de cualquier objeto que se haya escrito en el bucket mientras se aplicaba la configuración Aplicada al propietario del bucket. Estos objetos no pertenecen al escritor del objeto, incluso si vuelve a habilitar las ACL.

Para obtener instrucciones sobre cómo habilitar y administrar las ACL mediante la AWS Management Console, AWS Command Line Interface (CLI), la API de REST o los SDK de AWS, consulteConfiguración de la ACL.

Requisitos previos para desactivar las ACL

Antes de desactivar las ACL de un bucket existente, se deben completar los siguientes requisitos previos.

Revisión de las ACL de bucket y objeto y migración de los permisos de ACL

Cuando desactiva las ACL, los permisos otorgados por las ACL de bucket y objeto ya no afectan al acceso. Antes de desactivar las ACL, revise las ACL de bucket y objeto.

Si las ACL del bucket conceden permisos de lectura o escritura a otros que no pertenezcan a la cuenta, debe migrar estos permisos a la política de bucket para poder aplicar la configuración Aplicada al propietario del bucket. Si no migra las ACL de bucket que conceden acceso de lectura o escritura fuera de la cuenta, la solicitud de aplicar la configuración Aplicada al propietario del bucket genera un error y devuelve el código de error InvalidBucketAclWithObjectOwnership.

Por ejemplo, si desea desactivar las ACL de un bucket que recibe registros de acceso al servidor, debe migrar los permisos de ACL del bucket para el grupo de entrega de registros de S3 a la entidad principal del servicio de registro en una política de bucket. Para obtener más información, consulte Concesión de acceso al grupo de entrega de registros de S3 para el registro de acceso al servidor.

Si desea que el escritor del objeto mantenga el control total del objeto que carga, el escritor del objeto es la mejor configuración de propiedad de objetos para el caso de uso. Si desea controlar el acceso de objeto individual, la mejor opción es la configuración de propietario del bucket preferido. Estos casos de uso son poco frecuentes.

Para revisar las ACL y migrar los permisos de ACL a las políticas de bucket, consulte Requisitos previos para desactivar las ACL.

Identificar las solicitudes que requirieron una ACL para su autorización

Para identificar las solicitudes de Amazon S3 que requerían ACL para la autorización, puede utilizar el valor `aclRequired` de los registros de acceso al servidor de Amazon S3 o AWS CloudTrail.

Si la solicitud requería una ACL para su autorización o si tiene solicitudes PUT que especifican una ACL, la cadena es `Yes`. Si no se requerían ACL, si está estableciendo una ACL predefinida `bucket-owner-full-control` o si las solicitudes están permitidas por su política de buckets, la cadena de valor de `aclRequired` es "`-`"en los registros de acceso al servidor de Amazon S3 y falta en CloudTrail. Para obtener más información sobre los valores `aclRequired` previstos, consulte Valores de `aclRequired` para solicitudes comunes de Amazon S3.

Si tiene solicitudes `PutBucketAcl` o `PutObjectAcl` con encabezados que conceden permisos basados en ACL, con la excepción de la ACL predefinida `bucket-owner-full-control`, debe eliminar esos encabezados para poder desactivar las ACL. De lo contrario, sus solicitudes fallarán.

Para las demás solicitudes que requieran una ACL para la autorización, migre esos permisos de ACL a políticas de buckets. A continuación, elimine cualquier ACL de bucket antes de activar la configuración aplicada al propietario del bucket.

> **Note**
>
> No elimine las ACL de objetos. De lo contrario, las aplicaciones que dependen de las ACL de objetos para obtener permisos perderán el acceso.

Si ve que ninguna solicitud requiere una ACL para su autorización, puede proceder a desactivar las ACL. Para obtener más información sobre las solicitudes de identificación, consulte Uso de los registros de acceso al servidor de Amazon S3 para identificar solicitudes y Identificación de solicitudes de Amazon S3 mediante CloudTrail.

Revisión y actualización de las políticas de bucket que utilizan claves de condición relacionadas con la ACL

Después de aplicar la configuración Aplicada al propietario del bucket para desactivar las ACL, los nuevos objetos se pueden cargar en el bucket solo si la solicitud utiliza ACL de control total del propietario del bucket o no especifica una ACL. Antes de desactivar las ACL, revise la política de bucket para ver las claves de condición relacionadas con la ACL.

Si la política de bucket utiliza una clave de condición relacionada con la ACL para requerir la ACL predefinida `bucket-owner-full-control` (por ejemplo, `s3:x-amz-acl`), no es necesario actualizar la política de bucket. La siguiente política de bucket utiliza `s3:x-amz-acl` para requerir la ACL predefinida `bucket-owner-full-control` para solicitudes `PutObject` de S3. Esta política todavía requiere que el escritor del objeto especifique la ACL predefinida `bucket-owner-full-`

`control`. Sin embargo, los buckets con ACL desactivadas siguen aceptando esta ACL, por lo que las solicitudes se siguen realizando correctamente sin que se requieran cambios en el lado del cliente.

```
{
    "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:::amzn-s3-demo-bucket/*",
            "Condition": {
                "StringEquals": {
                    "s3:x-amz-acl": "bucket-owner-full-control"
                }
            }
        }
    ]
}
```

Sin embargo, si la política de bucket utiliza una clave de condición relacionada con la ACL que requiere una ACL diferente, debe quitar esta clave de condición. Esta política de bucket de ejemplo requiere la ACL `public-read` para solicitudes `PutObject` de S3 y, por lo tanto, se deben actualizar antes de desactivar las 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:::amzn-s3-demo-bucket/*",
            "Condition": {
                "StringEquals": {
                    "s3:x-amz-acl": "public-read"
                }
            }
        }
    ]
}
```

Permisos de propiedad de objetos

Para aplicar, actualizar o eliminar una configuración de propiedad de objetos de un bucket, es necesario el permiso `s3:PutBucketOwnershipControls`. Para devolver la configuración de propiedad de objetos de un bucket, es necesario el permiso `s3:GetBucketOwnershipControls`. Para obtener más información, consulte Configuración de la propiedad de objetos al crear un bucket y Visualización de la configuración de propiedad de objetos para un bucket de S3.

Desactivación de las ACL para todos los buckets nuevos

De forma predeterminada, todos los buckets nuevos se crean con la configuración Aplicada al propietario del bucket y con las ACL deshabilitadas. Recomendamos mantener las ACL deshabilitadas. Como regla general, se recomienda utilizar políticas basadas en recursos de S3 (políticas de bucket y políticas de punto de acceso) o políticas de IAM para el control de acceso en lugar de ACL. Las políticas son una opción de control de acceso simplificada y más flexible. Con las políticas de bucket y las políticas de puntos de acceso, puede definir reglas que se apliquen ampliamente a todas las solicitudes a sus recursos de Amazon S3.

Replicación y propiedad de objetos

Cuando utiliza la replicación de S3 y los buckets de origen y destino pertenecen a diferentes Cuentas de AWS, puede desactivar las ACL (con la configuración Aplicada al propietario del bucket de Propietario del objeto) para cambiar la propiedad de réplica a la Cuenta de AWS que posee el bucket de destino. Esta configuración imita el comportamiento de anulación del propietario existente sin necesidad del permiso `s3:ObjectOwnerOverrideToBucketOwner`. Todos los objetos que se replican en el bucket de destino con la configuración Aplicada al propietario del bucket pertenecen al

propietario del bucket de destino. Para obtener más información acerca de la opción de invalidación del propietario para configuraciones de replicación, consulte Cambiar el propietario de la réplica.

Configuración de la propiedad de objetos

Puede aplicar una configuración de propiedad de objetos mediante la consola de Amazon S3, AWS CLI, los SDK de AWS, la API de REST de Amazon S3 o AWS CloudFormation. La siguiente API de REST y los comandos de la AWS CLI admiten la propiedad de objetos:

API de REST	AWS CLI	Descripción
PutBucketOwnershipControls	put-bucket-ownership-controls	Crea o modifica la configuración de propiedad de objetos de un bucket de S3 existente.
CreateBucket	create-bucket	Crea un bucket utilizando el encabezado de solicitud de `x-amz-object-ownership` para especificar la configuración de propiedad de objetos.
GetBucketOwnershipControls	get-bucket-ownership-controls	Recupera la configuración de propiedad de objetos de un bucket de Amazon S3.
DeleteBucketOwnershipControls	delete-bucket-ownership-controls	Elimina la configuración de propiedad de objetos de un bucket de Amazon S3.

Para obtener más información acerca de la aplicación y el trabajo con la configuración de propiedad de objetos, consulte los siguientes temas.

Temas

- Requisitos previos para desactivar las ACL
- Configuración de la propiedad de objetos al crear un bucket
- Configuración de la propiedad de objetos en un bucket existente
- Visualización de la configuración de propiedad de objetos para un bucket de S3

- Desactivación de las ACL para todos los buckets nuevos y aplicación de la propiedad de objetos
- Resolución de problemas

Requisitos previos para desactivar las ACL

Si la ACL del bucket concede acceso fuera de la Cuenta de AWS, antes de desactivar las ACL, debe migrar los permisos de ACL del bucket a la política de bucket y restablecer la ACL del bucket a la ACL privada predeterminada. Si no migra estas ACL de bucket, la solicitud de aplicar la configuración Aplicada al propietario del bucket para desactivar las ACL genera un error y devuelve el código de error InvalidBucketAclWithObjectOwnership. También le recomendamos que revise los permisos de la ACL de objetos y los migre a la política de bucket. Para obtener más información acerca de otros requisitos previos sugeridos, consulte Requisitos previos para desactivar las ACL.

Cada una de las ACL de bucket y objeto existentes tiene un equivalente en una política de IAM. Los siguientes ejemplos de políticas de bucket muestran cómo los permisos `READ` y `WRITE` para las ACL de bucket y objeto se asignan a los permisos de IAM. Para obtener más información acerca de cómo se traduce cada ACL en permisos de IAM, consulte Mapeo de permisos de ACL y permisos de política de acceso.

Para revisar y migrar los permisos de ACL a las políticas de bucket, consulte los siguientes temas.

Temas

- Ejemplos de políticas de bucket
- Uso de la consola de S3 para revisar y migrar permisos de ACL
- Uso de la AWS CLI para revisar y migrar permisos de ACL
- Tutoriales de ejemplo

Ejemplos de políticas de bucket

En estas políticas de bucket de ejemplo, se muestra cómo migrar los permisos `READ` y `WRITE` de las ACL de bucket y objeto para una Cuenta de AWS de terceros a una política de bucket. Las ACL `READ_ACP` y `WRITE_ACP` son menos relevantes para las políticas porque conceden permisos relacionados con ACL (`s3:GetBucketAcl`, `s3:GetObjectAcl`, `s3:PutBucketAcl` y `s3:PutObjectAcl`).

Example : ACL **READ** para un bucket

Si su bucket tenía una ACL READ que concede a la Cuenta de AWS *111122223333* permiso para mostrar el contenido de su bucket, puede escribir una política de buckets que conceda los permisos `s3:ListBucket`, `s3:ListBucketVersions` y `s3:ListBucketMultipartUploads` para su bucket.

```
{
    "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 : ACL **READ** para todos los objetos de un bucket

Si cada objeto del bucket tiene una ACL READ que concede acceso a la Cuenta de AWS *111122223333*, puede escribir una política de buckets que conceda permisos `s3:GetObject` y `s3:GetObjectVersion` a esta cuenta para cada objeto del bucket.

```
{
    "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/*"
        }
    ]
}
```

Este elemento de recurso de ejemplo concede acceso a un objeto específico.

```
"Resource": "arn:aws:s3:::DOC-EXAMPLE-BUCKET/OBJECT-KEY"
```

Example : ACL **`WRITE`** que concede permisos para escribir objetos en un bucket

Si el bucket tiene una ACL `WRITE` que concede a Cuenta de AWS *111122223333* permiso para escribir objetos en el bucket, puede escribir una política de buckets que conceda permiso `s3:PutObject` para el bucket.

```
{
    "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/*"
        }
    ]
}
```

Uso de la consola de S3 para revisar y migrar permisos de ACL

Revisar permisos de ACL de un bucket

1. Inicie sesión AWS Management Console Management Console y abra la consola de Amazon S3 en https://console.aws.amazon.com/s3/.
2. En la lista de Buckets, seleccione el nombre del bucket.
3. Elija la pestaña Permisos.
4. En Access control list (ACL) (Lista de control de acceso [ACL]), revise los permisos de ACL del bucket.

Revisar permisos de ACL de un objeto

1. Inicie sesión AWS Management Console Management Console y abra la consola de Amazon S3 en https://console.aws.amazon.com/s3/.
2. En la lista Buckets, elija el nombre del bucket que contiene el objeto.
3. En la lista Objects (Objetos), elija el nombre del objeto.
4. Elija la pestaña Permisos.
5. En Access control list (ACL) (Lista de control de acceso [ACL]), revise los permisos de ACL del objeto.

Para migrar los permisos de ACL y actualizar la ACL del bucket

1. Inicie sesión AWS Management Console Management Console y abra la consola de Amazon S3 en https://console.aws.amazon.com/s3/.
2. En la lista de Buckets, seleccione el nombre del bucket.
3. En la pestaña Permissions (Permisos), en Bucket policy (Política de bucket), elija Edit (Editar).
4. En el cuadro Policy (Política), agregue o actualice la política de bucket.

   Para obtener ejemplos de políticas de bucket, consulte Ejemplos de políticas de bucket y Tutoriales de ejemplo.
5. Elija Guardar cambios.
6. Update your bucket ACL (Actualización de la ACL del bucket) para eliminar las concesiones de ACL a otros grupos o Cuentas de AWS.
7. Aplicar la opción Aplicada al propietario del bucket de Propiedad del objeto.

Uso de la AWS CLI para revisar y migrar permisos de ACL

1. Para devolver la ACL de bucket del bucket, utilice el comando get-bucket-acl de la AWS CLI:

```
aws s3api get-bucket-acl --bucket amzn-s3-demo-bucket
```

Por ejemplo, esta ACL de bucket concede acceso `WRITE` y `READ` a una cuenta de terceros. En esta ACL, la cuenta de terceros se identifica mediante el ID de usuario canónico. Para aplicar la configuración Aplicada al propietario del bucket y desactivar las ACL, debe migrar estos permisos para la cuenta de terceros a una política de bucket.

```
{
    "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"
        }
    ]
}
```

Para ver otras ACL de ejemplo, consulte Tutoriales de ejemplo.

2. Migración de los permisos de ACL del bucket a una política de bucket:

   En esta política de bucket de ejemplo, se concede permisos `s3:PutObject` y `s3:ListBucket` para una cuenta de terceros. En la política de bucket, la cuenta de terceros se identifica mediante el ID de la Cuenta de AWS (*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/*"
        }
    ]
}
```

   Para obtener más ejemplos de políticas de bucket, consulte Ejemplos de políticas de bucket y Tutoriales de ejemplo.

3. Para devolver la ACL de un objeto específico, utilice el comando get-object-acl de la AWS CLI.

```
aws s3api get-object-acl --bucket amzn-s3-demo-bucket --key EXAMPLE-OBJECT-KEY
```

4. Si es necesario, migre los permisos de ACL de objetos a la política de bucket.

   Este elemento de recurso de ejemplo concede acceso a un objeto específico de una política de bucket.

   ```
   "Resource": "arn:aws:s3:::amzn-s3-demo-bucket/EXAMPLE-OBJECT-KEY"
   ```

5. Restablezca la ACL del bucket a la ACL predeterminada.

   ```
   aws s3api put-bucket-acl --bucket amzn-s3-demo-bucket --acl private
   ```

6. Aplicar la opción Aplicada al propietario del bucket de Propiedad del objeto.

Tutoriales de ejemplo

En los siguientes ejemplos, se muestra cómo migrar los permisos de ACL a las políticas de bucket para casos de uso específicos.

Temas

- Concesión de acceso al grupo de entrega de registros de S3 para el registro de acceso al servidor
- Concesión de acceso público de lectura para los objetos de un bucket
- Concesión de acceso a Amazon ElastiCache (Redis OSS) al bucket de S3

Concesión de acceso al grupo de entrega de registros de S3 para el registro de acceso al servidor

Si desea aplicar la configuración Aplicada al propietario del bucket para desactivar los ACL de un bucket de destino de registro de acceso al servidor, debe migrar los permisos de ACL del bucket para el grupo de entrega de registros de S3 a la entidad principal del servicio de registro (`logging.s3.amazonaws.com`) en una política de bucket. Para obtener más información acerca de los permisos de entrega de registros, consulte Permisos para entrega de registros.

Esta ACL de bucket concede acceso `WRITE` y `READ_ACP` al grupo entrega de registros de 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"
        }
    ]
}
```

Para migrar los permisos de ACL del bucket para el grupo de entrega de registros de S3 a la entidad principal del servicio de registro en una política de bucket

1. Agregue la siguiente política de bucket al bucket de destino, sustituyendo los valores de ejemplo.

```
aws s3api put-bucket-policy --bucket amzn-s3-demo-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. Restablezca la ACL del bucket destino a la ACL predeterminada.

```
aws s3api put-bucket-acl --bucket amzn-s3-demo-bucket --acl private
```

3. Aplicar la opción Aplicada al propietario del bucket de Propiedad de objetos en el bucket de destino.

Concesión de acceso público de lectura para los objetos de un bucket

Si las ACL de objetos conceden acceso público de lectura a todos los objetos del bucket, puede migrar estos permisos de ACL a una política de bucket.

Esta ACL de objeto concede acceso público de lectura a un objeto de un bucket:

```
{
    "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"
        }
    ]
}
```

Para migrar los permisos de ACL de lectura pública a una política de bucket

1. Para conceder acceso público de lectura a todos los objetos del bucket, agregue la siguiente política de bucket y reemplace los valores de ejemplo.

```
aws s3api put-bucket-policy --bucket amzn-s3-demo-bucket --policy
 file://policy.json

policy.json:
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Sid": "PublicReadGetObject",
            "Effect": "Allow",
            "Principal": "*",
            "Action": [
                "s3:GetObject"
            ],
            "Resource": [
                "arn:aws:s3:::amzn-s3-demo-bucket/*"
            ]
        }
    ]
}
```

Para conceder acceso público a un objeto específico de una política de bucket, utilice el siguiente formato para el elemento `Resource`.

```
"Resource": "arn:aws:s3:::amzn-s3-demo-bucket/OBJECT-KEY"
```

Para conceder acceso público a todos los objetos que tengan un prefijo determinado, utilice el siguiente formato para el elemento `Resource`.

```
"Resource": "arn:aws:s3:::amzn-s3-demo-bucket/PREFIX/*"
```

2. Aplicar la opción Aplicada al propietario del bucket de Propiedad del objeto.

Concesión de acceso a Amazon ElastiCache (Redis OSS) al bucket de S3

Puede exportar la copia de seguridad de ElastiCache (Redis OSS) a un bucket de S3, lo que le permite tener acceso a la copia de seguridad desde fuera de ElastiCache. Para exportar la copia de seguridad a un bucket de S3, debe conceder permisos a ElastiCache para copiar una instantánea en el bucket. Si ha concedido permisos a ElastiCache en una ACL de bucket, debe migrar esos permisos a una política de bucket antes de aplicar la configuración de propietario del bucket obligatorio para desactivar las ACL. Para obtener más información, consulte Concesión de acceso a ElastiCache al bucket de Amazon S3 en la Guía del usuario de Amazon ElastiCache.

En el ejemplo siguiente se muestran los permisos de ACL del bucket que conceden permisos a 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"
        }
    ]
}
```

Migración de los permisos de ACL del bucket para ElastiCache (Redis OSS) a una política de bucket

1. Agregue la siguiente política de bucket al bucket, sustituyendo los valores de ejemplo.

```
aws s3api put-bucket-policy --bucket amzn-s3-demo-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:::amzn-s3-demo-bucket",
                "arn:aws:s3:::amzn-s3-demo-bucket/*"
            ]
        }
    ]
}
```

2. Restablezca la ACL del bucket a la ACL predeterminada:

```
aws s3api put-bucket-acl --bucket amzn-s3-demo-bucket --acl private
```

3. Aplicar la opción Aplicada al propietario del bucket de Propiedad del objeto.

Configuración de la propiedad de objetos al crear un bucket

Cuando se crea un bucket, es posible configurar S3 Object Ownership. Para configurar la propiedad de objetos de un bucket existente, consulte Configuración de la propiedad de objetos en un bucket existente.

S3 Object Ownership es una configuración de bucket de Amazon S3 que puede utilizar para desactivar las listas de control de acceso (ACL) y asumir la propiedad de todos los objetos del bucket, lo que simplifica la administración del acceso de los datos almacenados en Amazon S3. De forma predeterminada, S3 Object Ownership se establece en la configuración Aplicada al propietario del bucket. Además, las ACL están deshabilitadas para los buckets nuevos. Cuando las ACL están desactivadas, el propietario del bucket posee todos los objetos del bucket y administra el acceso a los datos de forma exclusiva mediante políticas de administración de acceso. Le recomendamos que mantenga las ACL desactivadas, excepto en circunstancias inusuales en las que necesite controlar el acceso a cada objeto de manera individual.

La propiedad de objetos tiene tres configuraciones que puede utilizar para controlar la propiedad de los objetos que se cargan en el bucket y desactivar o habilitar las ACL:

ACL desactivadas

- Propietario del bucket obligatorio (predeterminado): las ACL están desactivadas y el propietario del bucket tiene automáticamente la propiedad y el control total sobre cada objeto del bucket. Las ACL ya no afectan a los permisos de los datos del bucket de S3. El bucket utiliza políticas para definir el control de acceso.

ACL habilitadas

- Propietario del bucket preferido: el propietario del bucket tiene la propiedad y el control total sobre los nuevos objetos que otras cuentas escriben en el bucket con la ACL predefinida `bucket-owner-full-control`.
- Escritor del objeto: la Cuenta de AWS que carga un objeto es propietaria del objeto, tiene control total sobre él y puede conceder a otros usuarios acceso a él a través de ACL.

Permisos: para aplicar la configuración de Bucket owner enforced (Propietario del bucket obligatorio) o la configuración Bucket owner preferred (Propietario del bucket preferido), debe tener los siguientes permisos: `s3:CreateBucket` y `s3:PutBucketOwnershipControls`. No se necesitan permisos adicionales al crear un bucket con la configuración del Object writer (Escritor de objetos) aplicada. Para obtener más información sobre los permisos de Amazon S3, consulte Actions, resources, and condition keys for Amazon S3 en la Referencia de autorización de servicios.

> **Important**
>
> La mayoría de los casos de uso modernos de Amazon S3 ya no requieren el uso de ACL. Le recomendamos desactivar las ACL, excepto en circunstancias inusuales en las que necesite controlar el acceso de cada objeto de manera individual. Con la propiedad de objetos, puede desactivar las ACL y confiar en políticas para el control de acceso. Al desactivar las ACL, puede mantener fácilmente un bucket con objetos cargados por diferentes cuentas de AWS. Como propietario del bucket, posee todos los objetos del bucket y puede administrar el acceso a ellos mediante políticas.

Uso de la consola de S3

1. Inicie sesión en la AWS Management Console y abra la consola de Amazon S3 en https://console.aws.amazon.com/s3/.
2. En la barra de navegación de la parte superior de la página, elija el nombre de la Región de AWS que aparece. A continuación, elija en Region (Región) la región en la que desea crear un bucket.

   > **Note**
   >
   > Puede seleccionar una región cercana para minimizar la latencia y los costos, así como para satisfacer los requisitos normativos. Los objetos almacenados en una región nunca abandonarán esa región salvo que usted los transfiera de forma específica a otra. Para una lista de Regiones de AWS de Amazon S3, consulte Puntos de conexión de Servicio de AWS en la Referencia general de Amazon Web Services.

3. En el panel de navegación izquierdo, elija Instancias.
4. Elija Crear bucket.

   Se abrirá la página Crear bucket.
5. En Configuración general, vea la Región de AWS donde se creará el bucket.
6. En Tipo de depósito, seleccione Uso general.
7. En Nombre del bucket, escriba un nombre para el bucket.

   El nombre del bucket debe:

   - Ser exclusivo dentro de una partición. Una partición es una agrupación de regiones. AWS actualmente tiene tres particiones: `aws` (regiones estándar), `aws-cn` (regiones de China) y `aws-us-gov` (AWS GovCloud (US) Regions).
   - Tener entre 3 y 63 caracteres.
   - Consistir únicamente de letras minúsculas, números, puntos (.) y guiones (-). Para obtener una mejor compatibilidad, se recomienda evitar el uso de puntos (.) en los nombres de los buckets, excepto para los buckets que se utilizan únicamente para el alojamiento estático de sitios web.
   - Comenzar y terminar por un número o una letra.

Una vez que haya creado el bucket, no podrá modificar su nombre. Para obtener más información sobre la nomenclatura de los buckets, consulte Reglas de nomenclatura de buckets.

> **Important**
>
> Evite incluir información confidencial, como números de cuenta, en el nombre del bucket. El nombre del bucket será visible en las URL que señalan a los objetos almacenados en él.

8. AWS Management Console le permite copiar la configuración de un bucket existente en el nuevo bucket. Si no desea copiar la configuración de un bucket existente, vaya al paso siguiente.

> **Note**
>
> Esta opción:
>
> - No está disponible en la AWS CLI y solo está disponible en la consola
> - No está disponible para buckets de directorio
> - No copia la política de bucket del bucket existente al nuevo bucket

Para copiar la configuración de un bucket existente, en Copiar la configuración del depósito existente, seleccione Elegir bucket. Se abre la ventana Elegir bucket. Busque el bucket con los ajustes que quiera copiar y seleccione Elegir bucket. Se cierra la ventana Elegir bucket y se vuelve a abrir la ventana Crear bucket.

En Copiar la configuración del bucket existente, ahora verá el nombre del bucket que ha seleccionado. También verá la opción Restaurar los valores predeterminados que puede usar para eliminar la configuración del bucket copiada. Revise la configuración restante del bucket en la página Crear bucket. Verá que ahora coinciden con la configuración del bucket que seleccionó. Puede saltar al paso final.

9. En Propiedad de objetos, para desactivar o habilitar las ACL y controlar la propiedad de los objetos cargados en el bucket, elija una de las siguientes configuraciones:

ACL desactivadas

- Propietario del bucket obligatorio (predeterminado): las ACL están desactivadas y el propietario del bucket tiene automáticamente la propiedad y el control total sobre cada objeto del bucket. Las ACL ya no afectan a los permisos de acceso de los datos del bucket de S3. El bucket utiliza políticas exclusivamente para definir el control de acceso.

  Las ACL están desactivadas de forma predeterminada. La mayoría de los casos de uso modernos de Amazon S3 ya no requieren el uso de ACL. Le recomendamos que mantenga las ACL desactivadas, excepto en circunstancias inusuales en las que necesite controlar el acceso a cada objeto de manera individual. Para obtener más información, consulte Control de la propiedad de los objetos y desactivación de las ACL del bucket.

ACL habilitadas

- Propietario del bucket preferido: el propietario del bucket tiene la propiedad y el control total sobre los nuevos objetos que otras cuentas escriben en el bucket con la ACL predefinida `bucket-owner-full-control`.

  Si aplica la configuración de propietario del bucket preferido para requerir que todas las cargas de Amazon S3 incluyan la ACL predefinida `bucket-owner-full-control`, puede agregar una política de bucket que solo permita cargas de objetos que utilicen esta ACL.
- Escritor del objeto: la Cuenta de AWS que carga un objeto es propietaria del objeto, tiene control total sobre él y puede conceder a otros usuarios acceso a él a través de ACL.

> **Note**
>
> La configuración predeterminada es Aplicada al propietario del bucket. Para aplicar la configuración predeterminada y mantener las ACL deshabilitadas, solo se necesita el permiso `s3:CreateBucket`. Para habilitar las ACL, debe tener el permiso `s3:PutBucketOwnershipControls`.

10. En Configuración de bloqueo de acceso público para este bucket, elija la configuración Bloquear acceso público que desee aplicar al bucket.

    De forma predeterminada, las cuatro configuraciones de Bloqueo de acceso público estarán activas. Le recomendamos que deje todas las configuraciones activadas a menos que sepa

que necesita desactivar una o varias para su caso de uso específico. Para obtener más información acerca del bloqueo del acceso público, consulte Bloquear el acceso público a su almacenamiento de Amazon S3.

> **Note**
>
> Para habilitar todas las configuraciones de Bloqueo de acceso público, solo se requiere el permiso `s3:CreateBucket`. Para desactivar cualquier configuración de Bloqueo de acceso público, debe tener el permiso `s3:PutBucketPublicAccessBlock`.

11. (Opcional) En Control de versiones de buckets, puede elegir si desea mantener variantes de objetos en su bucket. Para obtener más información sobre el control de versiones, consulte Usar el control de versiones en buckets de S3.

    Para deshabilitar o habilitar el control de versiones en su bucket, elija Disable (Deshabilitar) o Enable (Habilitar).

12. (Opcional) En Tags (Etiquetas), puede elegir añadir etiquetas a su bucket. Las etiquetas son pares clave-valor que se utilizan para categorizar el almacenamiento de información.

    Para agregar una etiqueta de bucket, introduzca un valor en Clave y opcionalmente otro en Valor y elija Añadir etiqueta.

13. En Cifrado predeterminado, elija Editar.

14. Para configurar el cifrado predeterminado, en Tipo de cifrado, elija una de las siguientes opciones:

    - Clave administrada de Amazon S3 (SSE-S3)
    - Clave de AWS Key Management Service (SSE-KMS)

    > **Important**
    >
    > Si utiliza la opción de SSE-KMS para la configuración de cifrado predeterminado, se le aplicará la cuota de solicitudes por segundo (RPS) de AWS KMS. Para obtener más información acerca de las cuotas de AWS KMS y cómo solicitar un aumento de cuota, consulte Cuotas en la Guía para desarrolladores de AWS Key Management Service.

Los buckets y los objetos nuevos se cifran mediante el cifrado del lado del servidor con una clave administrada de Amazon S3 como nivel básico de configuración de cifrado. Para obtener más información acerca del cifrado predeterminado, consulte Establecer el comportamiento del cifrado predeterminado del lado del servidor para los buckets de Amazon S3.

Para obtener más información sobre el uso del cifrado del lado del servidor de Amazon S3 para cifrar los datos, consulte Uso del cifrado del servidor con claves administradas por Amazon S3 (SSE-S3).

15. Si ha elegido la clave de AWS Key Management Service (SSE-KMS), haga lo siguiente:

    a. En Clave de AWS KMS, especifique su clave de KMS de una de las siguientes maneras:

    - Para seleccionar de una lista de claves de KMS disponibles, marque Elija entre sus claves de AWS KMS keys y seleccione su clave de KMS de la lista de claves disponibles.

      En esta lista aparecen tanto la Clave administrada de AWS (`aws/s3`) como las claves administradas por el cliente. Para obtener más información acerca de las claves administradas por el cliente, consulte Claves de cliente y claves de AWS en la Guía para desarrolladores de AWS Key Management Service.
    - Para introducir el ARN de la clave de KMS, elija Introducir el ARN de la AWS KMS key e introduzca el ARN de la clave de KMS en el campo que aparece.
    - Para crear una nueva clave administrada por el cliente en la consola de AWS KMS, elija Crear una clave de KMS.

      Para obtener más información acerca de cómo crear una AWS KMS key, consulte Creación de claves en la AWS Key Management Service Guía para desarrolladores.

> **Important**
>
> Solo puede utilizar las claves de KMS que estén disponibles en la misma Región de AWS del bucket. La consola de Amazon S3 solo muestra las primeras 100 claves de KMS de la misma región del bucket. Para utilizar una clave de KMS que no aparezca en la lista, debe introducir el ARN de la clave de KMS. Si desea utilizar una clave de KMS propiedad de una cuenta de diferente, primero debe tener permiso para utilizar la clave y, después, debe introducir el ARN de la clave de KMS. Para obtener más información sobre los permisos entre cuentas para las

claves de KMS, consulte Crear claves de KMS que otras cuentas puedan utilizar en la Guía para desarrolladores de AWS Key Management Service. Para obtener más información sobre SSE-KMS, consulte Especificación del cifrado del lado del servidor con AWS KMS (SSE-KMS).

Cuando utilice una AWS KMS key para el cifrado en el lado del servidor en Amazon S3, debe elegir una clave de cifrado de KMS simétrica. Amazon S3 admite solo claves de KMS de cifrado simétricas y no claves de KMS asimétricas. Para obtener más información, consulte Identificación de claves de KMS simétricas y asimétricas en la Guía para desarrolladores de AWS Key Management Service.

Para obtener más información acerca de cómo crear una AWS KMS key, consulte Creación de claves en la AWS Key Management Service Guía para desarrolladores. Para obtener más información acerca del uso de AWS KMS con Amazon S3, consulte Uso del cifrado del servidor con claves de AWS KMS (SSE-KMS).

b. Cuando configure el bucket para que use el cifrado predeterminado con SSE-KMS, también puede habilitar las claves de bucket de S3. Las claves de bucket de S3 reducen el costo del cifrado al reducir el tráfico de solicitudes de Amazon S3 a AWS KMS. Para obtener más información, consulte Reducción del costo de SSE-KMS con las claves de bucket de Amazon S3.

   Para utilizar las claves de bucket de S3, en Clave de bucket, seleccione Habilitar.

16. (Opcional) Si desea habilitar el bloqueo de objetos en S3, haga lo siguiente:

a. Seleccione Advanced settings (Ajustes avanzados).

> **Important**
>
> Al habilitar Bloqueo de objetos, también se habilita el control de versiones para el bucket. Después de habilitar, debe configurar la retención predeterminada de Object Lock y la configuración de retención legal para evitar que los nuevos objetos se eliminen o se sobrescriban.

b. Si desea habilitar el bloqueo de objetos, elija Enable (Habilitar), lea la advertencia que aparece y acéptela.

Para obtener más información, consulte Usar Bloqueo de objetos de S3.

> **Note**
>
> Para crear un bucket con Bloqueo de objetos, debe tener los siguientes permisos: `s3:CreateBucket`, `s3:PutBucketVersioning` y `s3:PutBucketObjectLockConfiguration`.

17. Elija Crear bucket.

Uso de la AWS CLI

Para establecer la propiedad de objetos al crear un nuevo bucket, utilice el comando `create-bucket` de la AWS CLI con el parámetro `--object-ownership`.

En este ejemplo se aplica la configuración Aplicada al propietario del bucket para un nuevo bucket mediante la AWS CLI:

```
aws s3api create-bucket --bucket  amzn-s3-demo-bucket --region us-east-1 --object-
ownership BucketOwnerEnforced
```

> **Important**
>
> Si no configura la propiedad del objeto al crear un bucket mediante la AWS CLI, la configuración predeterminada será `ObjectWriter` (ACL habilitadas).

Uso de AWS SDK para Java

En este ejemplo se establece la configuración Aplicada al propietario del bucket para un nuevo bucket mediante 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);
```

Uso de AWS CloudFormation

Para utilizar el recurso `AWS::S3::Bucket` de AWS CloudFormation para configurar la propiedad de objetos al crear un bucket nuevo, consulte OwnershipControls en AWS::S3::Bucket en la Guía del usuario de AWS CloudFormation.

Uso de la API de REST

Para aplicar la configuración Aplicada al propietario del bucket de S3 Object Ownership, utilice la operación `CreateBucket` de la API con el encabezado de solicitud `x-amz-object-ownership` establecido en `BucketOwnerEnforced`. Para obtener información y ejemplos, consulte CreateBucket en la Referencia de la API de Amazon Simple Storage Service.

Pasos siguientes: Después de aplicar la configuración de propietario del bucket obligatorio o la configuración de propietario del bucket preferido de Propietario de objetos, puede seguir los siguientes pasos:

- Propietario del bucket obligatorio: requiere que todos los buckets nuevos se creen con ACL desactivadas mediante una política de IAM u Organizations.
- Propietario del bucket preferido: agrega una política de bucket de S3 para solicitar la ACL predefinida `bucket-owner-full-control` para todas las cargas de objetos en el bucket.

Configuración de la propiedad de objetos en un bucket existente

Puede configurar S3 Object Ownership en un bucket de S3 existente. Para aplicar la propiedad de objetos al crear un bucket, consulte Configuración de la propiedad de objetos al crear un bucket.

S3 Object Ownership es una configuración de bucket de Amazon S3 que puede utilizar para desactivar las listas de control de acceso (ACL) y asumir la propiedad de todos los objetos del bucket, lo que simplifica la administración del acceso de los datos almacenados en Amazon S3. De forma predeterminada, S3 Object Ownership se establece en la configuración Aplicada al propietario del bucket. Además, las ACL están deshabilitadas para los buckets nuevos. Cuando las ACL están desactivadas, el propietario del bucket posee todos los objetos del bucket y administra el acceso a los datos de forma exclusiva mediante políticas de administración de acceso. Le recomendamos que mantenga las ACL desactivadas, excepto en circunstancias inusuales en las que necesite controlar el acceso a cada objeto de manera individual.

La propiedad de objetos tiene tres configuraciones que puede utilizar para controlar la propiedad de los objetos que se cargan en el bucket y desactivar o habilitar las ACL:

ACL desactivadas

- Propietario del bucket obligatorio (predeterminado): las ACL están desactivadas y el propietario del bucket tiene automáticamente la propiedad y el control total sobre cada objeto del bucket. Las ACL ya no afectan a los permisos de los datos del bucket de S3. El bucket utiliza políticas para definir el control de acceso.

ACL habilitadas

- Propietario del bucket preferido: el propietario del bucket tiene la propiedad y el control total sobre los nuevos objetos que otras cuentas escriben en el bucket con la ACL predefinida `bucket-owner-full-control`.
- Escritor del objeto: la Cuenta de AWS que carga un objeto es propietaria del objeto, tiene control total sobre él y puede conceder a otros usuarios acceso a él a través de ACL.

Requisitos previos: antes de aplicar la configuración Aplicada al propietario del bucket para desactivar las ACL, debe migrar los permisos de ACL del bucket a las políticas de bucket y restablecer las ACL del bucket a la ACL privada predeterminada. También recomendamos que migre los permisos de ACL de objetos a las políticas de bucket y edite las políticas de bucket que requieren ACL distintas de las ACL de control total del propietario del bucket. Para obtener más información, consulte Requisitos previos para desactivar las ACL.

Permisos: para utilizar esta operación, debe tener el permiso `s3:PutBucketOwnershipControls`. Para obtener más información sobre los permisos de Amazon S3, consulte Actions, resources, and condition keys for Amazon S3 en la Referencia de autorización de servicios.

Uso de la consola de S3

1. Inicie sesión AWS Management Console Management Console y abra la consola de Amazon S3 en https://console.aws.amazon.com/s3/.
2. En la lista Buckets, elija el nombre del bucket para el que desea configurar S3 Object Ownership.
3. Elija la pestaña Permisos.
4. En Object Ownership (Propiedad de objeto), elija Edit (Editar).

5. En Object Ownership (Propiedad de objetos), para desactivar o habilitar las ACL y controlar la propiedad de los objetos cargados en el bucket, elija una de las siguientes configuraciones:

   ACL desactivadas

   - Propietario del bucket obligatorio: las ACL están desactivadas y el propietario del bucket tiene automáticamente la propiedad y el control total sobre cada objeto del bucket. Las ACL ya no afectan a los permisos de los datos del bucket de S3. El bucket utiliza políticas para definir el control de acceso.

     Para requerir que todos los buckets nuevos se creen con ACL desactivadas mediante políticas de IAM o AWS Organizations, consulte Desactivación de las ACL para todos los buckets nuevos (propietario del bucket obligatorio).

   ACL habilitadas

   - Propietario del bucket preferido: el propietario del bucket tiene la propiedad y el control total sobre los nuevos objetos que otras cuentas escriben en el bucket con la ACL predefinida `bucket-owner-full-control`.

     Si aplica la configuración de propietario del bucket preferido para requerir que todas las cargas de Amazon S3 incluyan la ACL predefinida `bucket-owner-full-control`, puede agregar una política de bucket que solo permita cargas de objetos que utilizan esta ACL.
   - Escritor del objeto: la Cuenta de AWS que carga un objeto es propietaria del objeto, tiene control total sobre él y puede conceder a otros usuarios acceso a él a través de ACL.
6. Seleccione Guardar.

Uso de la AWS CLI

Para aplicar una configuración de propiedad de objetos a un bucket existente, utilice el comando `put-bucket-ownership-controls` con el parámetro `--ownership-controls`. Los valores válidos de propiedad son `BucketOwnerEnforced`, `BucketOwnerPreferred` o `ObjectWriter`.

En este ejemplo se aplica la configuración Aplicada al propietario del bucket para un bucket existente mediante la AWS CLI:

```
aws s3api put-bucket-ownership-controls --bucket amzn-s3-demo-bucket --ownership-
controls="Rules=[{ObjectOwnership=BucketOwnerEnforced}]"
```

Para obtener más información sobre `put-bucket-ownership-controls`, consulte put-bucket-ownership-controls en la Guía del usuario de AWS Command Line Interface.

Uso de AWS SDK para Java

En este ejemplo se aplica la configuración `BucketOwnerEnforced` de Propiedad de objetos en un bucket existente mediante 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);
```

Uso de AWS CloudFormation

Para usar AWS CloudFormation para aplicar una configuración de propiedad de objetos para un bucket existente, consulte AWS::S3::Bucket OwnershipControls en la Guía del usuario de AWS CloudFormation.

Uso de la API de REST

Para utilizar la API de REST para aplicar una configuración de propiedad de objetos a un bucket de S3 existente, utilice `PutBucketOwnershipControls`. Para obtener más información, consulte PutBucketOwnershipControls en la Referencia de la API de Amazon Simple Storage Service.

Pasos siguientes: Después de aplicar la configuración de propietario del bucket obligatorio o la configuración de propietario del bucket preferido de Propietario de objetos, puede seguir los siguientes pasos:

- Propietario del bucket obligatorio: requiere que todos los buckets nuevos se creen con ACL desactivadas mediante una política de IAM u Organizations.
- Propietario del bucket preferido: agrega una política de bucket de S3 para solicitar la ACL predefinida `bucket-owner-full-control` para todas las cargas de objetos en el bucket.

Visualización de la configuración de propiedad de objetos para un bucket de S3

S3 Object Ownership es una configuración de bucket de Amazon S3 que puede utilizar para desactivar las listas de control de acceso (ACL) y asumir la propiedad de todos los objetos del bucket, lo que simplifica la administración del acceso de los datos almacenados en Amazon S3. De forma predeterminada, S3 Object Ownership se establece en la configuración Aplicada al propietario del bucket. Además, las ACL están deshabilitadas para los buckets nuevos. Cuando las ACL están desactivadas, el propietario del bucket posee todos los objetos del bucket y administra el acceso a los datos de forma exclusiva mediante políticas de administración de acceso. Le recomendamos que mantenga las ACL desactivadas, excepto en circunstancias inusuales en las que necesite controlar el acceso a cada objeto de manera individual.

La propiedad de objetos tiene tres configuraciones que puede utilizar para controlar la propiedad de los objetos que se cargan en el bucket y desactivar o habilitar las ACL:

ACL desactivadas

- Propietario del bucket obligatorio (predeterminado): las ACL están desactivadas y el propietario del bucket tiene automáticamente la propiedad y el control total sobre cada objeto del bucket. Las ACL ya no afectan a los permisos de los datos del bucket de S3. El bucket utiliza políticas para definir el control de acceso.

ACL habilitadas

- Propietario del bucket preferido: el propietario del bucket tiene la propiedad y el control total sobre los nuevos objetos que otras cuentas escriben en el bucket con la ACL predefinida `bucket-owner-full-control`.
- Escritor del objeto: la Cuenta de AWS que carga un objeto es propietaria del objeto, tiene control total sobre él y puede conceder a otros usuarios acceso a él a través de ACL.

Puede ver la configuración de la S3 Object Ownership de un bucket de Amazon S3. Para configurar la propiedad de objetos de un bucket nuevo, consulte Configuración de la propiedad de objetos

al crear un bucket. Para configurar la propiedad de objetos de un bucket existente, consulte Configuración de la propiedad de objetos en un bucket existente.

Permisos: para utilizar esta operación, debe tener el permiso `s3:GetBucketOwnershipControls`. Para obtener más información sobre los permisos de Amazon S3, consulte Actions, resources, and condition keys for Amazon S3 en la Referencia de autorización de servicios.

Uso de la consola de S3

1. Inicie sesión en la AWS Management Console y abra la consola de Amazon S3 en https://console.aws.amazon.com/s3/.
2. En la lista Buckets, elija el nombre del bucket para el que desea configurar la propiedad de objetos.
3. Elija la pestaña Permisos.
4. En Object Ownership (Propiedad de objetos), puede ver la configuración de la propiedad de objetos del bucket.

Uso de la AWS CLI

Para recuperar la configuración de S3 Object Ownership de un bucket de S3, utilice el comando get-bucket-ownership-controls de la AWS CLI.

```
aws s3api get-bucket-ownership-controls --bucket amzn-s3-demo-bucket
```

Uso de la API de REST

Para recuperar la configuración de la propiedad de objetos de un bucket de S3, utilice la operación de la API `GetBucketOwnershipControls`. Para obtener más información, consulte GetBucketOwnershipControls.

Desactivación de las ACL para todos los buckets nuevos y aplicación de la propiedad de objetos

Le recomendamos que desactive las ACL en los buckets de Amazon S3. Para ello, aplique la configuración Aplicada al propietario del bucket de S3 Object Ownership. Al aplicar esta configuración, las ACL se desactivan y automáticamente tiene la propiedad y el control total sobre todos los objetos del bucket. Para requerir que todos los buckets nuevos se creen con ACL

desactivadas, use políticas de AWS Identity and Access Management (IAM) o políticas de control de servicios (SCP) de AWS Organizations, como se describe en la siguiente sección.

Para aplicar la propiedad de objetos para objetos nuevos sin desactivar las ACL, puede aplicar la configuración de propietario del bucket preferido. Al aplicar esta configuración, le recomendamos encarecidamente que actualice la política de bucket para requerir la ACL predefinida `bucket-owner-full-control` para todas las solicitudes PUT que se realicen al bucket. Asegúrese también de actualizar los clientes para que envíen la ACL predefinida `bucket-owner-full-control` a su bucket desde otras cuentas.

Temas

- Desactivación de las ACL para todos los buckets nuevos (propietario del bucket obligatorio)
- Requisito de la ACL predefinida bucket-owner-full-control para las operaciones PUT de Amazon S3 (propietario del bucket preferido)

Desactivación de las ACL para todos los buckets nuevos (propietario del bucket obligatorio)

En el siguiente ejemplo, la política de IAM deniega el permiso `s3:CreateBucket` para un usuario o rol de IAM específico a menos que se aplique la configuración Aplicada al propietario del bucket de Propiedad de objetos. El par clave-valor en el bloque `Condition` especifica `s3:x-amz-object-ownership` como su clave y la configuración `BucketOwnerEnforced` como su valor. En otras palabras, el usuario de IAM solo puede crear buckets si establece la configuración Aplicada al propietario del bucket de Propiedad de objetos y desactiva las ACL. También puede utilizar esta política como SCP límite para la organización AWS.

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Sid": "RequireBucketOwnerFullControl",
            "Action": "s3:CreateBucket",
            "Effect": "Deny",
            "Resource": "*",
            "Condition": {
                "StringNotEquals": {
                    "s3:x-amz-object-ownership": "BucketOwnerEnforced"
                }
            }
        }
```

```
    ]
}
```

Requisito de la ACL predefinida bucket-owner-full-control para las operaciones **PUT** de Amazon S3 (propietario del bucket preferido)

Con la configuración de propietario del bucket preferido de Propiedad de objetos, como propietario del bucket tiene la propiedad y el control total sobre los objetos nuevos que otras cuentas escriben en el bucket con la ACL predefinida `bucket-owner-full-control`. Sin embargo, si otras cuentas escriben objetos en el bucket sin la ACL predefinida `bucket-owner-full-control`, el escritor del objeto mantiene el acceso de control total. Como propietario del bucket, puede implementar una política de bucket que permita escrituras solo si especifican la ACL predefinida `bucket-owner-full-control`.

> **Note**
>
> Si tiene las ACL desactivadas con la configuración Aplicada al propietario del bucket, como propietario del bucket tiene automáticamente la propiedad y el control total sobre todos los objetos del bucket. No es necesario utilizar esta sección para actualizar la política de bucket a fin de aplicar la propiedad de objetos para el propietario del bucket.

La siguiente directiva de bucket especifica que la cuenta *111122223333* puede cargar objetos *amzn-s3-demo-bucket* solo cuando la ACL del objeto está establecida en `bucket-owner-full-control`. Asegúrese de reemplazar *111122223333* con la cuenta y *amzn-s3-demo-bucket* con el nombre del bucket.

```
{
   "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:::amzn-s3-demo-bucket/*",
         "Condition": {
            "StringEquals": {
               "s3:x-amz-acl": "bucket-owner-full-control"
            }
         }
      }
   ]
}
```

A continuación, se muestra una operación de copia de ejemplo que incluye la ACL predefinida `bucket-owner-full-control` mediante la AWS Command Line Interface (AWS CLI).

```
aws s3 cp file.txt s3://amzn-s3-demo-bucket --acl bucket-owner-full-control
```

Una vez que se aplica la política de bucket, si el cliente no incluye la ACL predefinida `bucket-owner-full-control`, la operación produce un error y el cargador recibe el siguiente error:

An error occurred (AccessDenied) when calling the PutObject operation: Access Denied (Se ha producido un error [AccessDenied] al llamar a la operación PutObject: Acceso denegado).

> **Note**
>
> Si los clientes necesitan acceso a objetos después de la carga, tendrá que conceder permisos adicionales a la cuenta de carga. Para obtener información sobre cómo conceder acceso a las cuentas a los recursos, consulte Explicaciones que utilizan políticas para administrar el acceso a los recursos de Amazon S3.

Resolución de problemas

Cuando aplica la configuración de propietario del bucket obligatorio de S3 Object Ownership, las listas de control de acceso (ACL) se desactivan y, como propietario del bucket, automáticamente tiene la propiedad de todos los objetos del bucket. Las ACL ya no afectan a los permisos de los objetos del bucket. Puede utilizar políticas para conceder permisos. Todas las solicitudes PUT de S3 deben especificar la ACL `bucket-owner-full-control` predefinida o no especificar ninguna ACL, pues si no estas solicitudes fallarán. Para obtener más información, consulte Control de la propiedad de los objetos y desactivación de las ACL del bucket.

Si se especifica una ACL no válida o los permisos de ACL del bucket conceden acceso fuera de la Cuenta de AWS, es posible que vea las siguientes respuestas de error.

AccessControlListNotSupported

Después de aplicar la configuración Aplicada al propietario del bucket de Propiedad de objetos, las ACL se desactivan. Las solicitudes de configuración o actualización de ACL fallan con un error `400` y devuelven el código de error AccessControlListNotSupported. Las solicitudes de lectura de ACL siguen siendo compatibles. Las solicitudes de lectura de ACL siempre devuelven una respuesta que muestra el control total del propietario del bucket. En las operaciones PUT, debe especificar las ACL de control total del propietario del bucket o no especificar una ACL. De lo contrario, se producirá un error en las operaciones PUT.

El siguiente comando de ejemplo `put-object` AWS CLI incluye la ACL predefinida `public-read`.

```
aws s3api put-object --bucket amzn-s3-demo-bucket --key object-key-name --body doc-example-body --acl public-read
```

Si el bucket utiliza la configuración Aplicada al propietario del bucket para desactivar las ACL, esta operación genera un error y el cargador recibe el siguiente mensaje de error:

An error occurred (AccessControlListNotSupported) when calling the PutObject operation: The bucket does not allow ACLs (Se produjo un error (AccessControlListNotSupported) al llamar a la operación PutObject: el bucket no permite ACL)

InvalidBucketAclWithObjectOwnership

Si desea aplicar la configuración Aplicada al propietario del bucket para desactivar las ACL, la ACL del bucket debe otorgar control total únicamente al propietario del bucket. La ACL del bucket no puede dar acceso a una Cuenta de AWS externa ni a otro grupo. Por ejemplo, si la solicitud `CreateBucket` establece Aplicada al propietario del bucket y especifica una ACL de bucket que proporciona acceso a una Cuenta de AWS externa, se produce un error `400` en la solicitud y se devuelve el código de error InvalidBucketAclWithObjectOwnership. Del mismo modo, si la solicitud `PutBucketOwnershipControls` establece Aplicada al propietario del bucket en un bucket que tiene una ACL de bucket que concede permisos a otros, la solicitud falla.

Example : ACL de bucket existente concede acceso público de lectura

Por ejemplo, si una ACL de bucket existente concede acceso público de lectura, no se puede aplicar la configuración Aplicada al propietario del bucket de Propiedad de objetos hasta que migre

estos permisos de ACL a una política de bucket y restablezca la ACL del bucket a la ACL privada predeterminada. Para obtener más información, consulte Requisitos previos para desactivar las ACL.

En este ejemplo, la ACL de bucket concede acceso público de lectura:

```
{
    "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"
        }
    ]
}
```

El siguiente comando de ejemplo `put-bucket-ownership-controls` de la AWS CLI aplica la configuración Aplicada al propietario del bucket para Propiedad de objetos:

```
aws s3api put-bucket-ownership-controls --bucket amzn-s3-demo-bucket --ownership-
controls Rules=[{ObjectOwnership=BucketOwnerEnforced}]
```

Dado que la ACL del bucket concede acceso público de lectura, la solicitud falla y devuelve el siguiente código de error:

An error occurred (InvalidBucketAclWithObjectOwnership) when calling the PutBucketOwnershipControls operation: Bucket cannot have ACLs set with ObjectOwnership's BucketOwnerEnforced setting (Se produjo un error (InvalidBucketAclWithObjectOwnership) al llamar

a la operación PutBucketOwnershipControls: el bucket no puede tener ACL con la configuración ObjectOwnership's BucketOwnerEnforced)

Registro y monitoreo en Amazon S3

El monitoreo es una parte importante a la hora de mantener la fiabilidad, la disponibilidad y el rendimiento de Amazon S3 y las soluciones de AWS. Debe recopilar datos de monitoreo de todas las partes de su solución de AWS para que pueda depurar un error de múltiples puntos de una forma más fácil en caso de que se produzca dicho error. AWS proporciona varias herramientas para monitorear sus recursos de Amazon S3 y responder a posibles incidentes.

Para obtener más información, consulte Monitorización de Amazon S3.

> **Note**
>
> Para obtener más información sobre el uso de la clase de almacenamiento Amazon S3 Express One Zone con buckets de directorio, consulte ¿Qué es S3 Express One Zone? y Buckets de directorio.

Alarmas de Amazon CloudWatch

Con las alarmas de Amazon CloudWatch, puede ver una métrica determinada durante el periodo especificado. Si la métrica supera un límite determinado, se envía una notificación a un tema de Amazon SNS o a una política de AWS Auto Scaling. Las alarmas de CloudWatch no invocan acciones simplemente porque se encuentren en determinado estado. En su lugar, el estado debe haber cambiado y debe mantenerse durante el número de periodos especificado. Para obtener más información, consulte Monitorización de métricas con Amazon CloudWatch.

Registros de AWS CloudTrail

CloudTrail proporciona un registro de las medidas adoptadas por un usuario, un rol o un servicio de AWS en Amazon S3. Mediante la información recopilada por CloudTrail, puede determinar la solicitud que se realizó a Amazon S3, la dirección IP de origen desde la que se realizó, quién la realizó y cuándo, etc. Para obtener más información, consulte Registro de llamadas a la API de Amazon S3 mediante AWS CloudTrail.

Amazon GuardDuty

Amazon GuardDuty es un servicio de detección de amenazas que supervisa de forma continua las cuentas, los contenedores, las cargas de trabajo y los datos dentro del entorno de AWS para identificar posibles amenazas o riesgos de seguridad para los buckets de S3. GuardDuty también proporciona un contexto detallado sobre las amenazas que detecta. GuardDuty

supervisa los registros de administración de AWS CloudTrail en busca de amenazas y muestra información relevante para la seguridad. Por ejemplo, GuardDuty incluirá factores de una solicitud de API, como el usuario que realizó la solicitud, la ubicación desde la que se hizo la solicitud y la API específica solicitada, que podrían ser inusuales en el entorno. Protección de S3 en GuardDuty supervisa los eventos de datos de S3 recopilados por CloudTrail e identifica posibles comportamientos anómalos y maliciosos en todos los buckets de S3 del entorno.

Registros de acceso de Amazon S3

Los registros de acceso del servidor proporcionan registros detallados sobre las solicitudes que se realizan a un bucket. Los registros de acceso al servidor resultan útiles para muchas aplicaciones. Por ejemplo, la información del registro de acceso puede ser útil en auditorías de acceso y seguridad. Para obtener más información, consulte Registro de solicitudes con registro de acceso al servidor.

AWS Trusted Advisor

Trusted Advisor aprovecha las prácticas recomendadas aprendidas al atender a cientos de miles de clientes de AWS. Trusted Advisor inspecciona su entorno de AWS y realiza recomendaciones cuando surge la oportunidad de ahorrar dinero, mejorar el rendimiento y la disponibilidad del sistema o ayudar a cerrar deficiencias de seguridad. Todos los clientes de AWS tienen acceso a cinco comprobaciones de Trusted Advisor. Los clientes con un plan de soporte Business o Enterprise pueden ver todas las comprobaciones de Trusted Advisor.

Trusted Advisor cuenta con las siguientes comprobaciones relacionadas con Amazon S3:

- Configuración de registro de buckets de Amazon S3.
- Comprobaciones de seguridad de los buckets de Amazon S3 que tienen permisos de acceso abierto.
- Comprobaciones de la tolerancia a errores de los buckets de Amazon S3 que no tienen activado el control de versiones, o que lo tienen suspendido.

Para obtener más información, consulte AWS Trusted Advisor en la Guía del usuario de AWS Support.

Las siguientes prácticas recomendadas sobre seguridad también evalúan el registro y la monitorización:

- Identify and audit all your Amazon S3 buckets
- Implement monitoring using Amazon Web Services monitoring tools

- Habilitar AWS Config
- Enable Amazon S3 server access logging
- Use CloudTrail
- Monitor Amazon Web Services security advisories

Validación de la conformidad para Amazon S3

Auditores externos evalúan la seguridad y la conformidad de Amazon S3 como parte de varios programas de conformidad de AWS, incluidos los siguientes:

- Controles del Sistema y Organizaciones (System and Organization Controls, SOC)
- La norma de seguridad de datos del sector de pagos con tarjeta (PCI DSS)
- Programa Federal de Administración de Riesgos y Autorizaciones (Federal Risk and Authorization Management Program, FedRAMP)
- Ley de Portabilidad y Responsabilidad de Seguros Médicos de EE. UU (Health Insurance Portability and Accountability Act, HIPAA).

AWS proporciona una lista actualizada frecuentemente de los servicios de AWS adscritos al ámbito de los programas de conformidad en Servicios de AWS en el ámbito del programa de conformidad.

Los informes de auditoría de terceros están disponibles para su descarga mediante AWS Artifact. Para obtener más información, consulte Descarga de informes en AWS Artifact.

Para obtener más información acerca de los programas de conformidad de AWS, consulte Programas de conformidad de AWS.

Su responsabilidad de conformidad al utilizar Amazon S3 se determina en función de la sensibilidad de los datos, los objetivos de conformidad de la organización, así como de la legislación y los reglamentos aplicables. Si su uso de Amazon S3 está sujeto a conformidad con ciertos estándares, como HIPAA, PCI o FedRAMP, AWS proporciona recursos de ayuda:

- En las Guías de inicio rápido de seguridad y conformidad, se incluyen consideraciones sobre arquitectura y se ofrecen pasos para implementar los entornos de referencia centrados en la seguridad y la conformidad en AWS.
- En el documento Diseño de arquitecturas para la conformidad y la seguridad de HIPAA, se describe la manera en que las empresas utilizan AWS para poder cumplir los requisitos de HIPAA.
- Los recursos de conformidad de AWS proporcionan diferentes cuadernos de trabajo y guías que es posible que se apliquen a su sector y ubicación.
- AWS Config se puede utilizar para evaluar en qué medida las configuraciones de los recursos cumplen las prácticas internas, las directrices del sector y las normativas.
- AWS Security Hub ofrece una vista integral de su estado de seguridad en AWS que lo ayuda a comprobar la conformidad con las normas del sector de seguridad y las prácticas recomendadas.

- Usar Bloqueo de objetos de S3 puede ayudar a cumplir con los requisitos técnicos de los reguladores de servicios financieros (como SEC, FINRA y CFTC) que requieren un almacenamiento de datos Write Once, Read Many (WORM, Escritura única y lectura múltiple) para determinados tipos de libros e información de registros.
- Inventario de Amazon S3 puede ayudarle a auditar e informar sobre el estado de replicación y cifrado de los objetos para sus necesidades empresariales, de conformidad y legales.

Resiliencia en Amazon S3

La infraestructura global de AWS se divide en regiones y zonas de disponibilidad. Las Regiones de AWS proporcionan varias zonas de disponibilidad físicamente independientes y aisladas que se encuentran conectadas mediante redes con un alto nivel de rendimiento y redundancia, además de baja latencia. Estas zonas de disponibilidad ofrecen un medio eficaz de diseñar y utilizar aplicaciones y bases de datos. Tienen una mayor disponibilidad, tolerancia a errores y escalabilidad que las infraestructuras de centros de datos tradicionales únicos o múltiples. Si específicamente necesita reproducir sus datos en distancias geográficas mayores, puede utilizar Información general de la replicación de objetos, que permite la copia automática y asincrónica de objetos en buckets de diferentes Regiones de AWS.

Cada Región de AWS cuenta con varias zonas de disponibilidad. Puede implementar sus aplicaciones en varias zonas de disponibilidad en la misma región para la tolerancia de errores y la baja latencia. Las zonas de disponibilidad están conectadas entre sí con redes de fibra óptica rápidas y privadas, lo que permite diseñar aplicaciones con facilidad que conmuten por error entre las zonas de disponibilidad sin interrupciones.

Para obtener más información sobre las Regiones de AWS y las zonas de disponibilidad, consulte AWSInfraestructura global de .

Además de la infraestructura global de AWS, Amazon S3 ofrece varias características que lo ayudan con sus necesidades de resiliencia y copia de seguridad de los datos.

Configuración del ciclo de vida

La configuración del ciclo de vida es un conjunto de reglas que definen acciones que Amazon S3 aplica a un grupo de objetos. Con las reglas de configuración del ciclo de vida, puede indicarle a Amazon S3 que pase los objetos a otras clases de almacenamiento más económicas, que los archive o que los elimine. Para obtener más información, consulte Administración del ciclo de vida del almacenamiento.

Control de versiones

El control de versiones es una forma de conservar diversas variantes de un objeto en el mismo bucket. Puede utilizar el control de versiones para conservar, recuperar y restaurar todas las versiones de los objetos almacenados en su bucket de Amazon S3. Con el control de versiones, puede recuperarse fácilmente de acciones no deseadas del usuario y de errores de la aplicación. Para obtener más información, consulte Usar el control de versiones en buckets de S3.

Bloqueo de objetos de S3

Puede usar Bloqueo de objetos de S3 para almacenar objetos mediante un modelo de escritura única y lectura múltiple (WORM). Con Bloqueo de objetos de S3 puede evitar que se elimine o se sobrescriba un objeto durante un periodo de tiempo determinado o de manera indefinida. Bloqueo de objetos de S3 le permite cumplir con los requisitos normativos que precisen de almacenamiento WORM o agregar una capa adicional de protección frente a cambios y eliminaciones de objetos. Para obtener más información, consulte Usar Bloqueo de objetos de S3.

Clases de almacenamiento

Amazon S3 ofrece una gama de clases de almacenamiento para elegir según los requisitos de la carga de trabajo. Las clases de almacenamiento S3 Standard-IA y S3 One Zone-IA están diseñadas para datos a los que se accede aproximadamente una vez al mes y necesitan acceso en milisegundos. La clase de almacenamiento S3 Glacier Instant Retrieval está diseñada para datos de archivo de larga duración a los que se accede aproximadamente una vez por trimestre con acceso en milisegundos. Para los datos de archivo que no requieren acceso inmediato, como las copias de seguridad, puede utilizar las clases de almacenamiento S3 Glacier Flexible Retrieval o S3 Glacier Deep Archive. Para obtener más información, consulte Uso de las clases de almacenamiento de Amazon S3.

Las siguientes prácticas recomendadas sobre seguridad también evalúan la resiliencia:

- Enable versioning
- Consider Amazon S3 cross-region replication
- Identify and audit all your Amazon S3 buckets

Cifrado de copias de seguridad de Amazon S3

Si almacena copias de seguridad mediante Amazon S3, el cifrado de las copias de seguridad depende de la configuración de esos buckets. Amazon S3 proporciona un medio de definir el comportamiento de cifrado predeterminado para un bucket de S3. Puede configurar el cifrado predeterminado en un bucket para que todos los objetos se cifren cuando se almacenen en el bucket. El cifrado predeterminado es compatible con las claves almacenadas en AWS KMS (SSE-KMS). Para obtener más información, consulte Establecer el comportamiento del cifrado predeterminado del lado del servidor para los buckets de Amazon S3.

Para obtener más información sobre el control de versiones y el bloqueo de objetos, consulte los temas siguientes: Usar el control de versiones en buckets de S3 Usar Bloqueo de objetos de S3

Seguridad de la infraestructura en Amazon S3

Como se trata de un servicio administrado, Amazon S3 está protegido por los procedimientos de seguridad de red globales de AWS que se describen en el pilar de seguridad del AWS Well-Architected Framework.

El acceso a Amazon S3 a través de la red se realiza mediante las API publicadas por AWS. Los clientes deben admitir el protocolo de seguridad de la capa de transporte (TLS) 1.2. Le recomendamos que también admita TLS 1.3. (Para obtener más información sobre esta recomendación, consulte Conexiones más rápidas a la nube de AWS con TLS 1.3 en el Blog de seguridad de AWS). Los clientes también deben admitir conjuntos de cifrado con confidencialidad directa total (PFS) tales como Ephemeral Diffie-Hellman (DHE) o Elliptic Curve Diffie-Hellman Ephemeral (ECDHE). Además, las solicitudes se deben firmar con AWS Signature V4 o AWS Signature V2, lo que requiere que se proporcionen credenciales válidas.

Estas API se pueden invocar desde cualquier ubicación de red. No obstante, Amazon S3 admite políticas de acceso basadas en recursos, que pueden incluir restricciones en función de la dirección IP de origen. Puede utilizar las políticas de bucket de Amazon S3 para controlar el acceso a los buckets desde puntos de enlace específicos de Virtual Private Cloud (VPC) o VPC específicas. Este proceso aísla con eficacia el acceso de red a un bucket de Amazon S3 determinado únicamente desde la VPC específica de la red de AWS. Para obtener más información, consulte Control del acceso desde puntos de enlace de la VPC con políticas de bucket.

Las siguientes prácticas recomendadas sobre seguridad también abordan la seguridad de la infraestructura en Amazon S3:

- Consider VPC endpoints for Amazon S3 access
- Identify and audit all your Amazon S3 buckets

Configuración y análisis de vulnerabilidades en CM de Amazon S3

AWS gestiona las tareas de seguridad básicas, como la aplicación de parches en la base de datos y el sistema operativo (SO) de invitado, la configuración del firewall y la recuperación de desastres. Estos procedimientos han sido revisados y certificados por los terceros pertinentes. Para obtener más detalles, consulte los siguientes recursos de :

- Validación de la conformidad para Amazon S3
- Modelo de responsabilidad compartida
- Amazon Web Services: Overview of Security Processes

Las siguientes prácticas recomendadas sobre seguridad también evalúan la configuración y los análisis de vulnerabilidades en Amazon S3:

- Identify and audit all your Amazon S3 buckets
- Habilitar AWS Config

Prácticas recomendadas de seguridad para Amazon S3

Amazon S3 proporciona una serie de características de seguridad que debe tener en cuenta a la hora de desarrollar e implementar sus propias políticas de seguridad. Las siguientes prácticas recomendadas son directrices generales y no suponen una solución de seguridad completa. Puesto que es posible que estas prácticas recomendadas no sean adecuadas o suficientes para su entorno, considérelas como recomendaciones útiles en lugar de como normas.

Temas

Prácticas recomendadas de seguridad para Amazon S3

Las siguientes prácticas recomendadas para Amazon S3 pueden serle de utilidad para evitar incidentes de seguridad.

Desactivar Listas de control de acceso (ACL)

S3 Object Ownership es una configuración de nivel de bucket de Amazon S3 que puede usar para controlar la propiedad de los objetos que se cargan en el bucket y para activar o desactivar las ACL. De forma predeterminada, la propiedad de objetos se establece en la configuración impuesta por el propietario del bucket. Además, todas las ACL están deshabilitadas. Cuando las ACL están deshabilitadas, el propietario del bucket posee todos los objetos del bucket y administra el acceso a los datos de forma exclusiva mediante políticas de administración de acceso.

La mayoría de los casos de uso modernos de Amazon S3 ya no requieren utilizar listas de control de acceso (ACL). Le recomendamos desactivar las ACL, excepto en circunstancias inusuales en las que necesite controlar el acceso a cada objeto de manera individual. Para desactivar las ACL y tomar posesión de cada objeto del bucket, aplique la configuración de propietario del bucket obligatorio para la S3 Object Ownership. Al desactivar las ACL, puede mantener fácilmente un bucket con objetos cargados por diferentes Cuentas de AWS.

Cuando las ACL están desactivadas, el control de acceso de los datos se basa en políticas, como las siguientes:

- Políticas de usuario de AWS Identity and Access Management (IAM)

- Políticas de buckets de S3
- Políticas de punto de conexión de nube privada virtual (VPC)
- Políticas de control de servicios (SCP) de AWS Organizations

La desactivación de las ACL simplifica la administración de permisos y la auditoría. Las ACL están deshabilitadas de forma predeterminada para los buckets nuevos. También existe la posibilidad de desactivar las ACL para los buckets existentes. Si dispone de un bucket que ya tiene objetos, después de desactivar las ACL, las ACL de objeto y bucket ya no formarán parte del proceso de evaluación de acceso. En cambio, el acceso se concede o deniega sobre la base de políticas.

Antes de desactivar las ACL, asegúrese de hacer lo siguiente:

- Revise la política de bucket para asegurarse de que cubra todas las formas en que pretende conceder acceso al bucket fuera de la cuenta.
- Restablezca la ACL del bucket a la opción predeterminada (control total para el propietario del bucket).

Tras deshabilitar las ACL, se producen los siguientes comportamientos:

- Su bucket solo acepta solicitudes PUT que no especifiquen una ACL o solicitudes PUT con las ACL de control total del propietario del bucket. Estas ACL incluyen la ACL `bucket-owner-full-control` preconfigurada o formas equivalentes de esta ACL que se expresan en XML.
- Las aplicaciones existentes que admiten ACL de control total del propietario del bucket no se ven afectadas.
- Las solicitudes PUT que contienen otras ACL (por ejemplo, concesiones personalizadas a determinadas Cuentas de AWS) fallan y devuelven un error `400 (Bad Request)` con el código de error `AccessControlListNotSupported`.

Para obtener más información, consulte Control de la propiedad de los objetos y desactivación de las ACL del bucket.

Asegúrese que sus buckets de Amazon S3 empleen las políticas correctas y que no sean accesibles públicamente.

A menos que necesite de forma explícita que alguien en Internet pueda leer o escribir en su bucket de S3, debe asegurarse de que no sea público. Estos son algunos de los pasos que puede realizar para bloquear el acceso público:

- Use S3 Block Public Access. S3 Block Public Access le permite configurar fácilmente controles centralizados para limitar el acceso público a sus recursos de Amazon S3. Estos controles centralizados se aplican independientemente de cómo se creen los recursos. Para obtener más información, consulte Bloquear el acceso público a su almacenamiento de Amazon S3.
- Identifique las políticas de bucket de Amazon S3 que permiten una identidad comodín, como `"Principal": "*"` (que en realidad significa «cualquiera»). Busque también políticas que permitan una acción comodín `"*"` (que, efectivamente, permita al usuario llevar a cabo cualquier acción en el bucket de Amazon S3).
- Del mismo modo, busque listas de control de acceso (ACL) del bucket de Amazon S3 que permitan leer, escribir o concedan acceso completo a “Todos” o “Cualquier usuario de AWS autenticado”.
- Utilice la operación de la API `ListBuckets` para escanear todos los buckets de Amazon S3. A continuación, utilice `GetBucketAcl`, `GetBucketWebsite` y `GetBucketPolicy` para determinar si cada bucket cuenta con configuración y controles de acceso compatibles.
- Utilice AWS Trusted Advisor para inspeccionar la implementación de Amazon S3.
- Considere implementar controles de detección continuos mediante el uso de las Reglas de AWS Config administradas s3-bucket-public-read-prohibited y s3-bucket-public-write-prohibited.

Para obtener más información, consulte Administración de identidades y accesos para Amazon S3.

Identificación de las posibles amenazas para los buckets de Amazon S3 mediante Amazon GuardDuty

Amazon GuardDuty es un servicio de detección de amenazas que identifica amenazas potenciales a las cuentas, los contenedores, las cargas de trabajo y los datos del entorno de AWS. Mediante modelos de machine learning (ML) y capacidades de detección de anomalías y amenazas, Amazon GuardDuty supervisa continuamente los diferentes orígenes de datos para identificar y priorizar los posibles riesgos de seguridad y actividades maliciosas en el entorno. Al habilitar GuardDuty, ofrece detección de amenazas para los orígenes de datos fundamentales, que incluyen eventos de administración de AWS CloudTrail, registros de flujo de VPC y registros de DNS. Para extender la detección de amenazas a los eventos del plano de datos en buckets de S3, puede habilitar la característica Protección de S3 en GuardDuty. Esta característica detecta amenazas como la exfiltración de datos y el acceso sospechoso a los buckets de S3 a través de los nodos Tor. GuardDuty también establece un patrón de referencia normal en el entorno y, cuando identifica un comportamiento potencialmente anómalo, proporciona

información contextual para ayudarlo a corregir el bucket de S3 potencialmente comprometido o las credenciales de AWS. Para obtener más información, consulte GuardDuty.

Implementación del acceso a los privilegios mínimos

Cuando concede permisos, debe decidir a quién concede cada permiso y para qué recurso de Amazon S3 se lo concede. Habilite las acciones específicas que desea permitir en dichos recursos. Por lo tanto, recomendamos que conceda únicamente los permisos obligatorios para realizar una tarea. La implementación del acceso con privilegios mínimos es esencial a la hora de reducir los riesgos de seguridad y el impacto que podrían causar los errores o los intentos malintencionados.

Las siguientes herramientas están disponibles para implementar el acceso a los privilegios mínimos:

- Acciones de políticas para Amazon S3 y Límites de permisos para las entidades de IAM
- Cómo funciona Amazon S3 con IAM
- Información general de las Listas de control de acceso (ACL)
- Políticas de control de servicios

Para recibir asesoramiento sobre qué tener en cuenta al elegir uno o más de los mecanismos anteriores, consulte Administración de identidades y accesos para Amazon S3.

Uso de roles de IAM para aplicaciones y Servicios de AWS que requieren acceso a Amazon S3

Para que las aplicaciones que se ejecutan en Amazon EC2 u otros Servicios de AWS accedan a recursos de Amazon S3, deben incluir credenciales de AWS válidas en sus solicitudes a la API de AWS. Recomendamos no almacenar las credenciales de AWS de forma directa en la aplicación ni en una instancia de Amazon EC2. Estas son las credenciales a largo plazo que no rotan automáticamente y que podrían tener un impacto empresarial significativo si se comprometen.

En su lugar, utilice un rol de IAM para administrar temporalmente las credenciales para las aplicaciones o los servicios que necesiten acceder a Amazon S3. Cuando utiliza un rol, no tiene que distribuir credenciales a largo plazo (como un nombre de usuario y una contraseña o claves de acceso) a una instancia de Amazon EC2 o un servicio de Servicio de AWS como AWS Lambda. El rol proporciona permisos temporales que las aplicaciones pueden utilizar cuando hacen llamadas a otros recursos de AWS.

Para obtener más información, consulte los siguientes temas de la guía del usuario de IAM:

- Roles de IAM
- Situaciones habituales con los roles: usuarios, aplicaciones y servicios

Consideración del cifrado de datos en reposo

Dispone de las siguientes opciones para proteger datos en reposo en Amazon S3.

- Cifrado del lado del servidor: todos los buckets de Amazon S3 tienen el cifrado configurado de forma predeterminada y todos los objetos nuevos cargados en un bucket de S3 se cifran automáticamente en reposo. El cifrado del lado del servidor con claves administradas de Amazon S3 (SSE-S3) es la configuración de cifrado predeterminada para cada bucket de Amazon S3. Para usar otro tipo de cifrado, puede especificar el tipo de cifrado del servidor que se utilizará en las solicitudes PUT de S3 o puede establecer la configuración de cifrado predeterminada en el bucket de destino.

  Amazon S3 también proporciona estas opciones de cifrado del lado del servidor:

  - Cifrado del lado del servidor con claves AWS Key Management Service (AWS KMS) (SSE-KMS)
  - Cifrado del servidor con claves de AWS Key Management Service (AWS KMS) (SSE-KMS)
  - Cifrado en el servidor con claves proporcionadas por el cliente (SSE-C)

  Para obtener más información, consulte Protección de los datos con el cifrado del servidor.
- Cifrado del cliente: cifre datos del cliente y cargue los datos cifrados en Amazon S3. En este caso, administra el proceso de cifrado, las claves de cifrado y las herramientas relacionadas. Al igual que con el cifrado en el servidor, el cifrado en el cliente le puede ser de utilidad para minimizar el riesgo al cifrar los datos con una clave almacenada en un mecanismo distinto de que almacena los datos por sí mismo.

  Amazon S3 ofrece distintas opciones de cifrado en el cliente. Para obtener más información, consulte Protección de los datos con el cifrado del cliente.

Aplicación del cifrado de los datos en tránsito

Puede utilizar HTTPS (TLS) para ayudarle a evitar posibles ataques de acceso no autorizado o de manipulación del tráfico de red con ataques de «persona en medio» o similares. Le recomendamos permitir solo conexiones cifradas mediante HTTPS (TLS) utilizando la condición aws:SecureTransport en las políticas del bucket de Amazon S3.

> **Important**
>
> Le recomendamos que la aplicación no fije los certificados TLS de Amazon S3, ya que AWS no admite la fijación de certificados de confianza pública. S3 renueva

automáticamente los certificados y la renovación puede realizarse en cualquier momento antes de que venza el certificado. La renovación de un certificado genera un nuevo par de claves pública y privada. Si ha fijado un certificado de S3 que se ha renovado recientemente con una nueva clave pública, no podrá conectarse a S3 hasta que la aplicación utilice el nuevo certificado.

Considere también implementar controles de detección continuos mediante el uso de la regla AWS Config administrada s3-bucket-ssl-requests-only.

Consideración del uso del bloqueo de objetos de S3

El bloqueo de objetos de S3 permite almacenar objetos con un modelo de escritura única y lectura múltiple (WORM). El bloqueo de objetos de S3 le puede ser de utilidad para evitar la eliminación accidental o inadecuada de datos. Por ejemplo, puede utilizar el bloqueo de objetos de S3 para proteger los registros de AWS CloudTrail.

Para obtener más información, consulte Usar Bloqueo de objetos de S3.

Activación del control de versiones de S3

El control de versiones de S3 es una forma de conservar diversas variantes de un objeto en el mismo bucket. Puede utilizar el control de versiones para conservar, recuperar y restaurar todas las versiones de los objetos almacenados en su bucket de . Con el control de versiones, puede recuperarse fácilmente de acciones no deseadas del usuario y de errores de la aplicación.

Considere también implementar controles de detección continuos mediante el uso de la regla AWS Config administrada s3-bucket-versioning-enabled.

Para obtener más información, consulte Usar el control de versiones en buckets de S3.

Consideración del uso de la replicación entre regiones de S3

Aunque Amazon S3 almacena sus datos en diversas zonas de disponibilidad alejadas geográficamente, de forma predeterminada los requisitos de conformidad pueden exigir que almacene los datos incluso en ubicaciones aún más alejadas. La replicación entre regiones (CRR) de S3 le permite reproducir los datos entre Regiones de AWS alejadas para cumplir con estos requisitos. La CRR habilita la copia asincrónica y automática de los objetos entre buckets de diferentes Regiones de AWS. Para obtener más información, consulte Información general de la replicación de objetos.

> **Note**
>
> CRR precisa que tanto los buckets de S3 de origen como los de destino tengan habilitado el control de versiones.

Considere también implementar controles de detección continuos mediante el uso de la regla AWS Config administrada s3-bucket-replication-enabled.

Consideración de la posibilidad de utilizar puntos de conexión de VPC para acceder a Amazon S3

Un punto de conexión de la nube privada virtual (Virtual Private Cloud, VPC) de Amazon S3 es una entidad lógica dentro de una VPC que permite la conectividad solo a Amazon S3. Los puntos de conexión de VPC pueden ayudar a evitar que el tráfico pase por el Internet abierto.

Los puntos de enlace de VPC para Amazon S3 tienen dos formas de controlar el acceso a los datos de Amazon S3:

- Con las políticas de bucket de S3 puede controlar qué solicitudes, usuarios o grupos pueden acceder a través de un punto de conexión de VPC específico.
- Puede controlar que VPC o puntos de enlace de la VPC tienen acceso a sus buckets de S3 a través de las políticas de bucket de S3.
- Puede ser de utilidad para evitar la sustracción de datos mediante el uso de una VPC que no tiene una gateway de Internet.

Para obtener más información, consulte Control del acceso desde puntos de enlace de la VPC con políticas de bucket.

Uso de los servicios de seguridad AWS administrados para supervisar la seguridad de los datos

Varios servicios de seguridad AWS administrados pueden ayudarlo a identificar, evaluar y supervisar los riesgos de seguridad y cumplimiento de sus datos de Amazon S3. Estos servicios también pueden ayudarlo a proteger sus datos de esos riesgos. Estos servicios incluyen capacidades automatizadas de detección, monitorización y protección diseñadas para escalar desde los recursos de Amazon S3 para una sola Cuenta de AWS hasta los recursos para organizaciones que abarcan miles de cuentas.

Para obtener más información, consulte Monitorización de la seguridad de los datos con servicios de seguridad de AWS administrados.

Prácticas recomendadas de monitorización y auditoría de Amazon S3

Las siguientes prácticas recomendadas para Amazon S3 le pueden ser de utilidad para detectar los incidentes y los posibles puntos débiles de la seguridad.

Identificación y auditoría de todos los buckets de Amazon S3

La identificación de sus activos de TI es un aspecto fundamental de seguridad y control. Tiene que tener una visión de todos sus recursos de Amazon S3 para evaluar sus medidas de seguridad y tomar así las acciones pertinentes respecto a las posibles áreas débiles. Para auditar sus recursos, le recomendamos que haga lo siguiente:

- Utilice el editor de etiquetas para identificar los recursos que precisan más seguridad o una auditoría y utilice dichas etiquetas cuando tenga que buscarlos. Para obtener más información, consulte Buscar recursos para etiquetar en la Guía del usuario de etiquetado de recursos de AWS.
- Utilice el inventario de S3 para auditar e informar sobre el estado de replicación y cifrado de los objetos para sus necesidades empresariales, de conformidad y legales. Para obtener más información, consulte Inventario de Amazon S3.
- Cree grupos de recursos para sus recursos de Amazon S3. Para obtener más información, consulte ¿Qué son los grupos de recursos? en la Guía del usuario de AWS Resource Groups.

Implementación de la monitorización mediante las herramientas de supervisión de AWS

La monitorización es una parte importante del mantenimiento de la fiabilidad, la seguridad, la disponibilidad y el rendimiento de Amazon S3 y las soluciones de AWS. AWS brinda herramientas y servicios para ayudarlo a monitorizar Amazon S3 y los otros servicios de Servicios de AWS. Por ejemplo, puede monitorizar métricas de Amazon CloudWatch para Amazon S3, concretamente las métricas `PutRequests`, `GetRequests`, `4xxErrors` y `DeleteRequests`. Para obtener más información, consulte Monitorización de métricas con Amazon CloudWatch y Monitorización de Amazon S3.

Si desea ver un segundo ejemplo, consulte Ejemplo: Actividad del bucket de Amazon S3. Este ejemplo describe cómo crear una alarma de CloudWatch que se desencadena cuando se produce una llamada a la API de Amazon S3 para `PUT` o `DELETE` para una política de bucket, un ciclo de vida de bucket o una configuración de replicación de bucket, o para `PUT` una ACL de bucket.

Habilitación del registro de acceso al servidor de Amazon S3.

El registro de acceso al servidor brinda registros detallados de las solicitudes realizadas a un bucket. Los registros de acceso al servidor pueden ayudarle con la seguridad y la auditoría de accesos; le pueden ayudar a saber más sobre su base de clientes y a comprender su factura de Amazon S3. Para obtener información acerca de cómo activar el registro de acceso al servidor, consulte Registro de solicitudes con registro de acceso al servidor.

Considere también implementar controles de detección continuos mediante el uso de la regla AWS Config administrada s3-bucket-logging-enabled.

Usar AWS CloudTrail

AWS CloudTrail proporciona un registro de las medidas adoptadas por un usuario, un rol o un Servicio de AWS en Amazon S3. Puede utilizar la información recopilada por CloudTrail para determinar lo siguiente:

- La solicitud que se realizó a Amazon S3
- La dirección IP desde la que se realizó la solicitud
- Quién realizó la solicitud
- La hora a la que se realizó la solicitud
- Detalles adicionales sobre la solicitud

Por ejemplo, puede identificar entradas de CloudTrail para acciones que afecten al acceso a los datos, concretamente `PUT`, `PutBucketAcl`, `PutObjectAcl`, `PutBucketPolicy` y `PutBucketWebsite`.

Cuando se configura una Cuenta de AWS, CloudTrail se habilita de forma predeterminada. Puede ver los eventos recientes en la consola de CloudTrail. Para crear un registro continuo de actividad y eventos para los buckets de Amazon S3 puede crear un seguimiento en la consola de CloudTrail. Para obtener más información, consulte Registro de eventos de datos en la Guía del usuario de AWS CloudTrail.

Al crear un seguimiento, puede configurar CloudTrail para que registre los eventos de datos. Los eventos de datos son registros de operaciones de recursos realizadas en o dentro de un recurso. En Amazon S3, los eventos de datos registran la actividad de API en el nivel de objeto para buckets individuales. CloudTrail admite un subconjunto de operaciones de API en el nivel de objetos de Amazon S3, como `GetObject`, `DeleteObject` y `PutObject`. Para obtener más información acerca de cómo funciona CloudTrail con Amazon S3, consulte Registro de llamadas a la API de Amazon S3 mediante AWS CloudTrail. En la consola de Amazon S3, también puede

configurar los buckets de S3 en Habilitación del registro de eventos de CloudTrail para buckets y objetos de S3.

AWS Config proporciona una regla administrada (`cloudtrail-s3-dataevents-enabled`) que puede utilizar para confirmar que al menos un seguimiento de CloudTrail está registrando eventos de datos para los buckets de S3. Para obtener más información, consulte cloudtrail-s3-dataevents-enabled en la Guía para desarrolladores de AWS Config.

Habilitar AWS Config

Varias de las prácticas recomendadas que se enumeran en este tema sugieren la creación de reglas de AWS Config. AWS Config le permite examinar, auditar y evaluar las configuraciones de los recursos de AWS. AWS Config monitorea las configuraciones de los recursos para que pueda evaluar las configuraciones registradas frente a las configuraciones de seguridad deseadas. En AWS Config, tiene las siguientes opciones:

- Revisar los cambios en las configuraciones y las relaciones entre los recursos de AWS.
- Investigar los historiales detallados de la configuración de recursos.
- Determinar el cumplimiento general de las configuraciones especificadas en sus directrices internas.

Al usar AWS Config puede simplificar las auditorías de conformidad, los análisis de seguridad, la administración de cambios y la resolución de problemas operativos. Para obtener más información, consulte Configuración de AWS Config mediante la consola en la Guía para desarrolladores de AWS Config. Al especificar los tipos de recursos para registrar, asegúrese de incluir los recursos de Amazon S3.

> **Important**
>
> Las reglas administradas de AWS Config solo admiten buckets de uso general al evaluar los recursos de Amazon S3. AWS Config no registra los cambios de configuración de los buckets de directorio. Para obtener más información, consulte Reglas administradas de AWS Config y Lista de reglas administradas de AWS Config en la Guía para desarrolladores de AWS Config.

Para ver un ejemplo de cómo utilizar AWS Config, consulte la entrada sobre Cómo usar AWS Config Config para monitorizar y responder a los buckets de Amazon S3 al permitir el acceso público en el Blog de seguridad de AWS.

Descubrimiento de datos confidenciales mediante Amazon Macie

Amazon Macie es un servicio de seguridad que descubre información confidencial mediante el machine learning y la coincidencia de patrones. Macie proporciona visibilidad sobre los riesgos de seguridad de los datos y permite automatizar la protección contra esos riesgos. Con Macie, puede automatizar la detección y la notificación de información confidencial en su conjunto de datos de Amazon S3 para comprender mejor los datos que almacena su organización en S3.

Para detectar información confidencial con Macie, puede utilizar criterios y técnicas integradas diseñados para detectar una lista grande y que no para de crecer de tipos de datos confidenciales para muchos países y regiones. Estos tipos de datos confidenciales incluyen varios tipos de información de identificación personal (PII), datos financieros y datos de credenciales. También existe la posibilidad de utilizar criterios personalizados definidos por usted: expresiones regulares que definen patrones de texto para que coincidan y, opcionalmente, secuencias de caracteres y reglas de proximidad para refinar los resultados.

Si Macie detecta datos confidenciales en un objeto de S3, Macie genera un resultado de seguridad para notificárselo. Este resultado proporciona información sobre el objeto afectado, los tipos y el número de ocurrencias de los datos confidenciales que ha encontrado Macie y detalles adicionales que le ayudan a investigar el bucket y el objeto de S3 afectados. Para obtener más información, consulte la Guía del usuario de Amazon Macie.

Uso de la lente de almacenamiento de S3

Lente de almacenamiento de S3 es una función de análisis de almacenamiento en la nube que puede utilizar para obtener visibilidad en toda la organización sobre el uso y la actividad del almacenamiento de objetos. S3 Storage Lens analiza también las métricas para ofrecer recomendaciones contextuales que puede usar para optimizar los costos de almacenamiento y aplicar las prácticas recomendadas para proteger los datos.

Lente de almacenamiento de S3 permite usar métricas para generar información resumida, como averiguar cuánto almacenamiento tiene en toda la organización o cuáles son los buckets y los prefijos de crecimiento más rápido. También puede utilizar las métricas de Lente de almacenamiento de S3 para identificar oportunidades de optimización de costos, implementar las prácticas recomendadas de protección de datos y administración de acceso y mejorar el rendimiento de las cargas de trabajo de las aplicaciones.

Por ejemplo, puede identificar los buckets que no tienen reglas del ciclo de vida de S3 para que aborten las cargas multipartes incompletas que tengan más de 7 días de antigüedad. También puede identificar los buckets que no siguen las prácticas recomendadas de protección de

datos, como el uso de la Replicación de S3 o el control de versiones de S3. Para obtener más información, consulte Comprensión de Amazon S3 Storage Lens.

Supervisión de los avisos de seguridad de AWS

Le recomendamos que compruebe con regularidad los avisos sobre seguridad publicados en Trusted Advisor para su Cuenta de AWS. Tenga en cuenta sobre todo los avisos sobre los buckets de Amazon S3 con permisos de acceso abierto. Puede hacerlo mediante programación o a través de la describe-trusted-advisor-checks.

Además, monitorice de forma activa la dirección principal de correo electrónico registrada en cada una de sus Cuentas de AWS. AWS contactará con usted, a través de esta dirección de correo electrónico, para informarle sobre los problemas de seguridad que surjan y que pudieran afectarle.

Los problemas operativos de AWS con gran alcance se publican en AWS Health Dashboard Service Health. Los problemas operativos también se publican en las cuentas individuales a través de AWS Health Dashboard. Para obtener más información, consulte la Documentación de AWS Health.

Monitorización de la seguridad de los datos con servicios de seguridad de AWS administrados

Varios servicios de seguridad AWS administrados pueden ayudarlo a identificar, evaluar y supervisar los riesgos de seguridad y cumplimiento de sus datos de Amazon S3. También pueden ayudarlo a proteger sus datos de esos riesgos. Estos servicios incluyen capacidades automatizadas de detección, monitorización y protección diseñadas para escalar desde los recursos de Amazon S3 para una sola Cuenta de AWS hasta los recursos para organizaciones que abarcan miles de Cuentas de AWS.

Los servicios de detección y respuesta de AWS pueden ayudarlo a identificar posibles errores de configuración de seguridad, amenazas o comportamientos inesperados, de modo que pueda responder rápidamente a actividades potencialmente no autorizadas o malintencionadas en su entorno. Los servicios de protección de datos de AWS pueden ayudarlo a monitorizar y proteger sus datos, cuentas y cargas de trabajo del acceso no autorizado. También pueden ayudarlo a descubrir información confidencial, como información de identificación personal (PII), en su conjunto de datos de Amazon S3.

Para ayudarlo a identificar y evaluar los riesgos de seguridad y cumplimiento de los datos, los servicios de seguridad de AWS administrados generan resultados para notificarle los posibles eventos o problemas de seguridad con sus datos de Amazon S3. Los resultados proporcionan detalles relevantes que puede utilizar para investigar, evaluar y actuar ante estos riesgos de acuerdo con sus flujos de trabajo y políticas de respuesta a incidentes. Puede acceder a los datos de los resultados directamente desde cada servicio. También puede enviar los datos a otras aplicaciones, servicios y sistemas, como el sistema de gestión de incidentes y eventos de seguridad (SIEM).

Para monitorizar la seguridad de sus datos de Amazon S3, considere la posibilidad de utilizar estos servicios de seguridad de AWS administrados.

Amazon GuardDuty

Amazon GuardDuty es un servicio de detección de amenazas que monitoriza de forma continua sus cargas de trabajo y Cuentas de AWS para detectar actividades maliciosas y ofrece resultados detallados sobre la seguridad para mejorar la visibilidad y la corrección.

Con la función de protección de S3 de GuardDuty, puede configurar GuardDuty para que analice los eventos de administración y datos de AWS CloudTrail de sus recursos de Amazon S3. GuardDuty luego monitoriza esos eventos para detectar actividades maliciosas y sospechosas.

Para fundamentar el análisis e identificar los posibles riesgos de seguridad, GuardDuty utiliza fuentes de información sobre amenazas y el machine learning.

GuardDuty puede monitorizar diferentes tipos de actividad para sus recursos de Amazon S3. Por ejemplo, los eventos de administración de CloudTrail para Amazon S3 incluyen operaciones en el nivel de bucket como `ListBuckets`, `DeleteBucket` y `PutBucketReplication`. Los eventos de datos de CloudTrail para Amazon S3 incluyen operaciones en el nivel de objeto como `GetObject`, `ListObjects` y `PutObject`. Si GuardDuty detecta una actividad anómala o potencialmente maliciosa, genera un resultado para notificárselo.

Para obtener más información, consulte Protección de Amazon S3 en Amazon GuardDuty en la guía del usuario de Amazon GuardDuty.

Amazon Detective

Amazon Detective simplifica el proceso de investigación y lo ayuda a realizar investigaciones sobre la seguridad de forma más rápida y eficaz. Detective proporciona agregaciones de datos, resúmenes y contextos prediseñados que pueden ayudarlo a analizar y evaluar la naturaleza y el alcance de los posibles problemas de seguridad.

Detective extrae automáticamente los eventos temporales, como las llamadas a la API desde AWS CloudTrail y los registros de flujo de Amazon VPC para sus recursos de AWS. También recopila los resultados generados por Amazon GuardDuty. A continuación, Detective utiliza el machine learning, el análisis estadístico y la teoría de grafos para generar visualizaciones que lo ayuden a realizar investigaciones sobre la seguridad eficaces con mayor rapidez.

Estas visualizaciones proporcionan una vista unificada e interactiva del comportamiento de los recursos y de las interacciones entre ellos a lo largo del tiempo. Puede explorar este gráfico de comportamiento para examinar las acciones potencialmente malintencionadas, como los intentos fallidos de inicio de sesión o las llamadas sospechosas a la API. También puede ver cómo afectan estas acciones a los recursos, así como a los buckets y los objetos de S3.

Para obtener más información, consulte la Guía de administración de Amazon Detective.

Analizador de acceso de IAM

AWS Identity and Access Management Access Analyzer (Analizador de acceso de IAM) permite identificar los recursos que se comparten con una entidad externa. También puede usar Analizador de acceso de IAM para validar las políticas de IAM comparándolas con la gramática de las políticas y las prácticas recomendadas, y generar políticas de IAM basadas en la actividad de acceso de sus registros de AWS CloudTrail.

Analizador de acceso de IAM utiliza un razonamiento basado en la lógica para analizar las políticas de recursos de su entorno de AWS, como las políticas de bucket. Con Analizador de acceso de IAM para Amazon S3, se le avisa cuando un bucket de S3 se configura para permitir el acceso a cualquiera en Internet u otras Cuentas de AWS, incluidas las cuentas fuera de la organización. Por ejemplo, Analizador de acceso de IAM para S3 puede notificar que un bucket tiene acceso de lectura o escritura a través de una lista de control de acceso (ACL) de bucket, una política de bucket, una política de punto de acceso de varias regiones o una política de punto de acceso. Para cada bucket público o compartido, recibe resultados que le informan del origen y el nivel de acceso público o compartido. Con estos resultados, puede adoptar medidas correctivas inmediatas y precisas para restaurar el acceso al bucket según lo previsto.

Para obtener más información, consulte Revisión del acceso al bucket mediante Analizador de acceso de IAM para S3.

Amazon Macie

Amazon Macie es un servicio de seguridad de datos que descubre datos confidenciales mediante el machine learning y la coincidencia de patrones, proporciona visibilidad de los riesgos de seguridad de los datos y permite establecer una protección automatizada contra esos riesgos.

Con Macie, puede automatizar la detección y la notificación de información confidencial en sus buckets de S3 para comprender mejor los datos que almacena su organización en Amazon S3. Para detectar datos confidenciales, puede utilizar los criterios y técnicas integrados que proporciona Macie, los criterios personalizados que usted defina o una combinación de ambos. Si Macie detecta datos confidenciales en un objeto de S3, Macie genera un resultado para notificárselo. Este resultado proporciona información sobre el objeto y el bucket afectados, los tipos y el número de ocurrencias de los datos confidenciales que ha encontrado Macie y detalles adicionales que le permitirán investigar.

Macie también proporciona estadísticas y otros datos que ofrecen información sobre el estado de la seguridad de sus datos de Amazon S3 y evalúa y supervisa automáticamente sus buckets de S3 para ofrecer seguridad y control de acceso. Si Macie detecta un posible problema con la seguridad o la privacidad de sus datos, como un bucket de acceso público, Macie genera un resultado para que lo revise y solucione según sea necesario.

Para obtener más información, consulte la Guía del usuario de Amazon Macie.

AWS Security Hub

AWS Security Hub es un servicio de administración de la postura de seguridad que comprueba las prácticas recomendadas de seguridad, agrega alertas y resultados de múltiples fuentes en un solo formato y permite la corrección automática.

Security Hub recopila y proporciona datos de resultados de seguridad de las soluciones de seguridad de AWS Partner Network integradas y los Servicios de AWS, incluidas Amazon Detective, Amazon GuardDuty, IAM Access Analyzer y Amazon Macie. También genera sus propios resultados mediante la ejecución continua de controles de seguridad automatizados y continuos basados en las prácticas recomendadas de AWS y en los estándares del sector compatibles.

A continuación, Security Hub correlaciona y consolida los resultados entre los proveedores para que pueda priorizar los resultados más importantes. También proporciona soporte para acciones personalizadas, que puede utilizar para invocar respuestas o acciones de corrección para clases específicas de resultados.

Con Security Hub, puede evaluar el estado de seguridad y cumplimiento de sus recursos de Amazon S3, y puede hacerlo como parte de un análisis más amplio de la postura de seguridad de su organización en Regiones de AWS individuales y en varias regiones. Esto incluye analizar las tendencias de seguridad e identificar los problemas de seguridad de mayor prioridad. También puede agregar resultados de varias Regiones de AWS y monitorizar y procesar los datos de los resultados agregados de una sola región.

Para obtener más información, consulte Controles de Amazon Simple Storage Service en la Guía del usuario de AWS Security Hub.

Gestionar el almacenamiento de Amazon S3

Después de crear buckets y cargar objetos en Amazon S3, puede administrar el almacenamiento de objetos mediante características como el control de versiones, clases de almacenamiento, bloqueo de objetos, operaciones por lotes, replicación, etiquetas, etc. En las secciones siguientes se proporciona información detallada sobre las capacidades y características de gestión de almacenamiento disponibles en Amazon S3.

> **Note**
>
> Para obtener más información sobre el uso de la clase de almacenamiento Amazon S3 Express One Zone con buckets de directorio, consulte ¿Qué es S3 Express One Zone? y Buckets de directorio.

Temas

Usar el control de versiones en buckets de S3

El control de versiones de Amazon S3 es una forma de conservar diversas variantes de un objeto en el mismo bucket. Puede utilizar S3 Versioning para conservar, recuperar y restaurar todas las versiones de los objetos almacenados en sus buckets. Con el control de versiones, se puede recuperar fácilmente de acciones no deseadas del usuario y de errores de la aplicación. Luego de habilitar el control de versiones para un bucket, si Amazon S3 recibe varias solicitudes de escritura para el mismo objeto simultáneamente, almacena todos los objetos.

Los buckets con el control de versiones habilitado le permiten recuperar objetos ante su eliminación o sobrescritura accidental. Por ejemplo, si elimina un objeto, Amazon S3 inserta un marcador de eliminación en lugar de eliminarlo de forma permanente. El marcador de eliminación se convierte en la versión actual del objeto. Si sobrescribe un objeto, se creará una nueva versión del objeto en el bucket. Siempre puede restaurar la versión anterior. Para obtener más información, consulte Eliminar versiones de objetos de un bucket con control de versiones habilitado.

De forma predeterminada, S3 Versioning está deshabilitado en los buckets y debe habilitarlo explícitamente. Para obtener más información, consulte Habilitar el control de versiones en buckets.

> **Note**
>
> - La API SOAP no admite Control de versiones de S3. La compatibilidad con SOAP por HTTP está obsoleta, pero aún se encuentra disponible con HTTPS. Las características nuevas de Amazon S3 no son compatibles con SOAP.
> - Se aplican tasas normales de Amazon S3 por cada versión de un objeto almacenado y transferido. Cada versión de un objeto es el objeto en sí, no se limita a ser una diferenciación de la versión anterior. Por tanto, si tiene tres versiones de un objeto almacenado, se le cobrará por tres objetos.

Buckets sin control de versiones, habilitados para control de versiones y suspendidos para control de versiones

Los buckets pueden estar en uno de los tres estados:

- No versionado (predeterminado)
- Control de versiones: habilitado

- Control de versiones: suspendido

Habilita y suspende el control de versiones en el nivel de bucket. Tras habilitar el control de versiones en un bucket, nunca puede volver a un estado sin control de versiones. Pero puede suspender el control de versiones en ese bucket.

El estado del control de versiones se aplica a todos los objetos (nunca solo a una parte) del bucket. Cuando habilita el control de versiones en un bucket, todos los objetos nuevos tienen una versión y se les asigna un identificador de versión único. Los objetos que ya existían en el bucket en el momento en que se habilitó el control de versiones siempre tendrán una versión y se les asignará un identificador de versión único cuando sean modificados por futuras solicitudes. Tenga en cuenta lo siguiente:

- Los objetos que se almacenan en un bucket antes de establecer el estado del control de versiones tienen el ID de versión `null`. Al habilitar el control de versiones, los objetos existentes en el bucket no cambian. Lo que cambia es la forma en la que Amazon S3 administrará los objetos en las solicitudes futuras. Para obtener más información, consulte Trabajar con objetos en un bucket con control de versiones habilitado.
- El propietario del bucket (o cualquier usuario con los permisos adecuados) puede suspender el control de versiones para dejar de acumular versiones de objetos. Al suspender el control de versiones, los objetos existentes en el bucket no cambian. Lo que cambia es la forma en la que Amazon S3 administrará los objetos en las solicitudes futuras. Para obtener más información, consulte Trabajar con objetos en un bucket con control de versiones suspendido.

Uso de S3 Versioning con S3 Lifecycle

Para personalizar su enfoque de retención de datos y controlar los costos de almacenamiento, utilice el control de versiones de los objetos con S3 Lifecycle. Para obtener más información, consulte Administración del ciclo de vida del almacenamiento. Para obtener más información acerca de cómo crear configuraciones de ciclo de vida de S3 mediante la AWS Management Console, la AWS CLI, los SDK de AWS o la API de REST, consulte Configuración de un ciclo de vida en un bucket.

> **Important**
>
> Si cuenta con una configuración de ciclo de vida de vencimiento de objetos en el bucket sin control de versiones y quiere mantener el mismo comportamiento de eliminación cuando habilite el control de versiones, debe agregar una configuración de vencimiento no actual.

La configuración de vencimiento de ciclo de vida no actual administra las eliminaciones de las versiones de objetos no actuales en el bucket habilitado para el control de versiones. (Un bucket habilitado para el control de versiones mantiene una versión actual del objeto y cero o más versiones no actuales del objeto). Para obtener más información, consulte Configuración de un ciclo de vida en un bucket.

Para obtener información sobre cómo trabajar con S3 Versioning, consulte los siguientes temas.

Temas

- Cómo funciona S3 Versioning
- Habilitar el control de versiones en buckets
- Configurar la eliminación de MFA
- Trabajar con objetos en un bucket con control de versiones habilitado
- Trabajar con objetos en un bucket con control de versiones suspendido

Cómo funciona S3 Versioning

Puede utilizar S3 Versioning para mantener varias versiones de un objeto en un bucket para poder restaurar objetos que se eliminan o sobrescriben accidentalmente. Por ejemplo, si aplica el control de versiones de S3 a un bucket, se producen los siguientes cambios:

- Si elimina un objeto, en lugar de eliminarlo permanentemente, Amazon S3 inserta un marcador de eliminación, que se convierte en la versión del objeto actual. Luego, puede restaurar la versión anterior. Para obtener más información, consulte Eliminar versiones de objetos de un bucket con control de versiones habilitado.
- Si sobrescribe un objeto, Amazon S3 añade una nueva versión del objeto en el bucket. La versión anterior permanece en el bucket y pasa a ser una versión no actual. Puede restaurar la versión anterior.

> **Note**
>
> Se aplican tasas normales de Amazon S3 por cada versión de un objeto almacenado y transferido. Cada versión de un objeto es el objeto en sí, no se limita a ser una diferenciación

de la versión anterior. Por tanto, si tiene tres versiones de un objeto almacenado, se le cobrará por tres objetos.

Cada bucket de S3 que crea cuenta con un subrecurso de control de versiones asociado. (Para obtener más información, consulte Opciones de configuración de buckets.) De forma predeterminada, su bucket no tendrá control de versiones y, por tanto, el subrecurso de control de versiones almacena una configuración de control de versiones vacía.

```
<VersioningConfiguration xmlns="http://s3.amazonaws.com/doc/2006-03-01/">
</VersioningConfiguration>
```

Para habilitar el control de versiones, puede enviar una solicitud a Amazon S3 con una configuración de control de versiones que incluya un estado `Enabled`.

```
<VersioningConfiguration xmlns="http://s3.amazonaws.com/doc/2006-03-01/">
  <Status>Enabled</Status>
</VersioningConfiguration>
```

Para suspender el control de versiones, puede configurar el valor de estado como `Suspended`.

> **Note**
>
> Al habilitar el control de versiones en un bucket por primera vez, es posible que el cambio se propague por completo en un instante. Para emitir operaciones de escritura (`PUT` o `DELETE`) en los objetos del bucket, se recomienda que espere 15 minutos después de habilitar el control de versiones.

El propietario del bucket y todos los usuarios autorizados de AWS Identity and Access Management (IAM) pueden habilitar el control de versiones. El bucket es propiedad de la Cuenta de AWS que creó el bucket. Para obtener más información sobre los permisos, consulte Administración de identidades y accesos para Amazon S3.

Para obtener más información acerca de cómo habilitar y desactivar el control de versiones de S3 mediante la AWS Management Console, AWS Command Line Interface (AWS CLI) o la API de REST, consulte the section called “Habilitar el control de versiones en buckets”.

Temas

- ID de versión
- Flujos de trabajo del control de versiones

ID de versión

Si habilita el control de versiones para un bucket, Amazon S3 genera automáticamente un ID de versión único para el objeto que se almacena. En un bucket, por ejemplo, puede tener dos objetos con la misma clave (nombre de objeto) pero ID de versión diferentes, como `photo.gif` (versión 111111) y `photo.gif` (versión 121212).
Diagrama que muestra un bucket con control de versiones habilitado que tiene dos objetos con la misma clave pero diferentes ID de versión.

Cada objeto tiene un ID de versión, independientemente de si S3 Versioning está habilitado o no. Si no ha habilitado S3 Versioning, Amazon S3 configura el valor del ID de versión en `null`. Si S3 Versioning está activado, Amazon S3 asigna un valor de ID de versión para el objeto. Este valor distingue dicho objeto de otras versiones de la misma clave.

Cuando se habilita S3 Versioning en un bucket existente, los objetos que ya están almacenados en el bucket no se modifican. Sus ID de versión (`null`), el contenido y los permisos siguen siendo los mismos. Después de habilitar Control de versiones de S3, cada objeto que se agrega al bucket obtiene un ID de versión, que lo distingue de otras versiones de la misma clave.

Solo Amazon S3 genera ID de versión y no se pueden editar. Los ID de versión son cadenas opacas unicode, codificadas en UTF-8, listas para URL que no tienen más de 1024 bytes de longitud. A continuación se muestra un ejemplo:

```
3sL4kqtJlcpXroDTDmJ+rmSpXd3dIbrHY+MTRCxf3vjVBH40Nr8X8gdRQBpUMLUo
```

> **Note**
>
> Para mayor simplicidad, los otros ejemplos de este tema utilizan ID mucho más cortos.

Flujos de trabajo del control de versiones

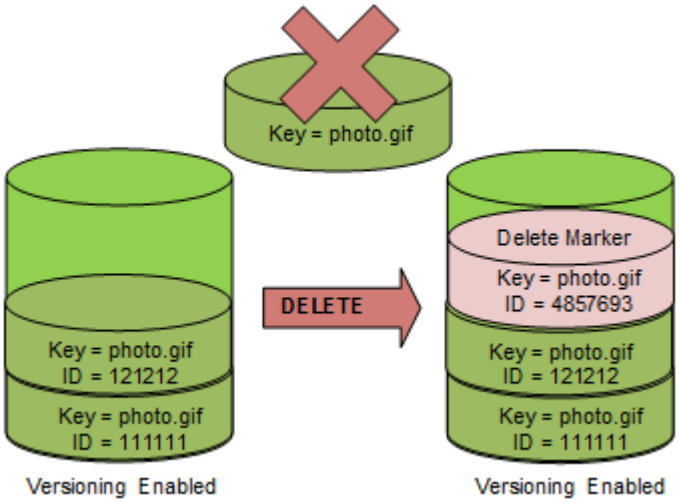
Cuando realiza una operación PUT para un objeto en un bucket con control de versiones habilitado, la versión no actual no se sobrescribirá. Como se muestra en el siguiente gráfico, cuando una nueva

versión de `photo.gif` se somete a una operación `PUT` en un bucket que ya contiene un objeto con el mismo nombre, se produce el siguiente comportamiento:

- El objeto original (ID = 111111) permanece en el bucket.
- Amazon S3 genera un nuevo identificador de versión (121212) y agrega esta versión más reciente del objeto al bucket.

Con esta funcionalidad, puede recuperar una versión anterior de un objeto si un objeto se ha sobrescrito o eliminado accidentalmente.

Cuando realiza una operación `DELETE` en un objeto, todas las versiones permanecen en el bucket y Amazon S3 inserta un marcador de eliminación, como se muestra en el siguiente gráfico.



El marcador de eliminación se convierte en la versión actual del objeto. De forma predeterminada, las solicitudes `GET` recuperarán la versión almacenada más recientemente. Realizar una solicitud `GET Object` cuando la versión actual es un marcador de eliminación devuelve un error `404 Not Found`, como se muestra en el siguiente gráfico.

Sin embargo, puede realizar una operación `GET` en una versión no actual de un objeto especificando su ID de versión. En el siguiente gráfico, se realiza una operación `GET` sobre una versión de objeto específica, 111111. Amazon S3 devuelve la versión del objeto aunque no sea la versión actual.

Para obtener más información, consulte Recuperar versiones de objetos de un bucket habilitado para el control de versiones.

Puede eliminar permanentemente un objeto especificando la versión que quiera eliminar. Solo el propietario de un bucket de Amazon S3 o un usuario de IAM autorizado puede eliminar

permanentemente una versión. Si la operación `DELETE` especifica el `versionId`, la versión del objeto se elimina permanentemente y Amazon S3 no inserta un marcador de eliminación.

Se pueden agregar factores adicionales de seguridad al configurar un bucket para habilitar la eliminación de autenticación multifactor (MFA). Cuando habilita la eliminación de MFA en un bucket, el propietario del bucket debe incluir dos formas de autenticación en cualquier solicitud para eliminar una versión o cambiar el estado de control de versiones del bucket. Para obtener más información, consulte Configurar la eliminación de MFA.

¿Cuándo se crean las nuevas versiones de un objeto?

Las nuevas versiones de objetos se crean solo cuando usted `PUT` un nuevo objeto. Tenga en cuenta que ciertas acciones como `CopyObject` funcionan mediante la implementación de una operación `PUT`.

Algunas acciones que modifican el objeto actual no crean una nueva versión ya que no `PUT` un objeto nuevo. Esto incluye acciones como cambiar las etiquetas de un objeto.

> **Important**
>
> Si detecta un aumento significativo en el número de respuestas de HTTP 503 (Servicio no disponible) recibidas para solicitudes `PUT` o `DELETE` de Amazon S3 a objetos en un bucket de Amazon S3 con Control de versiones de S3 habilitado, puede que tenga uno o varios objetos en el bucket para los que habrá millones de versiones. Para obtener más información, consulte la sección Control de versiones de S3 de Solución de problemas.

Habilitar el control de versiones en buckets

Utilice S3 Versioning para mantener varias versiones de un objeto en un bucket. En esta sección, se proporcionan ejemplos de cómo habilitar el control de versiones en un bucket mediante la consola, la API de REST, los SDK de AWS y AWS Command Line Interface (AWS CLI).

> **Note**
>
> Si habilita el control de versiones en un bucket por primera vez, es posible que el cambio tarde unos 15 minutos en propagarse. Para emitir operaciones de escritura (PUT o DELETE) en los objetos del bucket, se recomienda que espere 15 minutos después de habilitar el

control de versiones. Las operaciones de escritura realizadas antes de que se complete la conversión pueden aplicarse a objetos no versionados.

Para obtener más información sobre el control de versiones de S3, consulte Usar el control de versiones en buckets de S3. Para obtener información sobre cómo trabajar con objetos que se encuentran en buckets con control de versiones habilitado, consulte Trabajar con objetos en un bucket con control de versiones habilitado.

Para obtener más información sobre cómo utilizar el control de versiones de S3 para proteger los datos, consulte el Tutorial: Protecting data on Amazon S3 against accidental deletion or application bugs using S3 Versioning, S3 Object Lock, and S3 Replication (Protección de los datos en Amazon S3 contra la eliminación accidental o los errores en la aplicación mediante el control de versiones de S3, S3 Object Lock y la Replicación de S3).

Cada bucket de S3 que crea cuenta con un subrecurso de control de versiones asociado. (Para obtener más información, consulte Opciones de configuración de buckets.) De forma predeterminada, su bucket no tendrá control de versiones y, por tanto, el subrecurso de control de versiones almacena una configuración de control de versiones vacía.

```
<VersioningConfiguration xmlns="http://s3.amazonaws.com/doc/2006-03-01/">
</VersioningConfiguration>
```

Para habilitar el control de versiones, puede enviar una solicitud a Amazon S3 con una configuración de control de versiones que incluya un estado.

```
<VersioningConfiguration xmlns="http://s3.amazonaws.com/doc/2006-03-01/">
  <Status>Enabled</Status>
</VersioningConfiguration>
```

Para suspender el control de versiones, puede configurar el valor de estado como `Suspended`.

El propietario del bucket y todos los usuarios autorizados pueden habilitar el control de versiones. El propietario del bucket es la Cuenta de AWS que creó el bucket (la cuenta raíz). Para obtener más información sobre los permisos, consulte Administración de identidades y accesos para Amazon S3.

En las siguientes secciones, se proporcionan más detalles sobre cómo habilitar el control de versiones de S3 a través de la consola, la AWS CLI y los SDK de AWS.

Uso de la consola de S3

Siga estos pasos para utilizar la AWS Management Console a fin de habilitar el control de versiones en un bucket de S3.

Para habilitar o deshabilitar el control de versiones en un bucket de S3:

1. Inicie sesión en la AWS Management Console y abra la consola de Amazon S3 en https://console.aws.amazon.com/s3/.
2. En la lista Buckets (Buckets), elija el nombre del bucket para el que desea activar el control de versiones.
3. Seleccione Properties (Propiedades).
4. En Bucket Versioning (Control de versiones del bucket), elija Edit (Editar).
5. Elija Suspend (Suspender) o Enable (Habilitar) y, a continuación, elija Save changes (Guardar cambios).

> **Note**
>
> Puede utilizar la autenticación multifactor (MFA) de AWS con el control de versiones. Cuando utiliza MFA con el control de versiones, debe proporcionar las claves de acceso de su Cuenta de AWS y un código válido del dispositivo MFA de la cuenta para eliminar de manera permanente una versión de un objeto o suspender o volver a activar el control de versiones. Para utilizar la MFA con el control de versiones, habilite `MFA Delete`. Sin embargo, no puede habilitar `MFA Delete` mediante la AWS Management Console. Para ello debe utilizar AWS Command Line Interface (AWS CLI) o la API. Para obtener más información, consulte Configurar la eliminación de MFA.

Mediante AWS CLI

En el ejemplo siguiente se habilita el control de versiones en un bucket de S3.

```
aws s3api put-bucket-versioning --bucket amzn-s3-demo-bucket1 --versioning-
configuration Status=Enabled
```

En el siguiente ejemplo se habilita el control de versiones de S3 y la eliminación de autenticación multifactor (MFA) en un bucket.

```
aws s3api put-bucket-versioning --bucket amzn-s3-demo-bucket1 --versioning-
configuration Status=Enabled,MFADelete=Enabled --mfa "SERIAL 123456"
```

> **Note**
>
> El uso de la eliminación de MFA requiere un dispositivo de autenticación física o virtual aprobado. Para obtener más información sobre el uso de eliminación de MFA en Amazon S3, consulte Configurar la eliminación de MFA.

Para obtener más información sobre cómo habilitar el control de versiones mediante la AWS CLI, consulte put-bucket-versioning en la Referencia de comandos de AWS CLI.

Uso de los AWS SDK

Mediante los siguientes ejemplos, se habilita el control de versiones en un bucket y, a continuación, se recupera el estado de control de versiones mediante AWS SDK for Java y AWS SDK for .NET. Para obtener información acerca del uso de otros SDK de AWS, consulte el Centro de desarrolladores de AWS.

.NET

Para obtener información acerca de cómo configurar y ejecutar ejemplos de código, consulte Introducción al SDK de AWS para .NET en la Guía para desarrolladores del SDK de AWS para .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

Para obtener instrucciones sobre cómo crear y probar una muestra funcional, consulte Introducción en la Guía del desarrollador de AWS SDK for Java.

```
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

Con el siguiente ejemplo de código de Python, se crea un bucket de Amazon S3, se habilita el control de versiones y se configura un ciclo de vida que establece el vencimiento de las versiones de objetos no actuales después de 7 días.

```
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
```

Configurar la eliminación de MFA

Cuando se trabaja con S3 Versioning en buckets de Amazon S3, puede agregar de forma opcional otra capa de seguridad al configurar un bucket para habilitar la eliminación con MFA (autenticación multifactor). Si lo hace, el propietario del bucket debe incluir dos formas de autenticación en cualquier solicitud para eliminar una versión o cambiar el estado de control de versiones del bucket.

La eliminación de MFA precisa una autenticación adicional para cualquiera de las siguientes operaciones:

- Cambiar el estado de control de versiones del bucket
- Eliminar de forma permanente la versión de un objeto

La eliminación MFA requiere dos formas combinadas de autenticación:

- Sus credenciales de seguridad
- La concatenación de un número de serie válido, un espacio y el código de seis dígitos que se muestra en un dispositivo de autenticación autorizado

La eliminación de MFA refuerza la seguridad en caso de que, por ejemplo, sus credenciales de seguridad estén en riesgo. La eliminación de MFA puede ayudar a prevenir las eliminaciones accidentales de buckets ya que requiere que el usuario que inicia la acción de eliminación pruebe la posesión física de un dispositivo MFA con un código MFA y agregue una capa adicional de fricción y seguridad a la acción de eliminación.

Para identificar buckets que tienen habilitada la eliminación de MFA, puede utilizar las métricas de Lente de almacenamiento de S3. Lente de almacenamiento de S3 es una función de análisis de almacenamiento en la nube que puede utilizar para obtener visibilidad en toda la organización sobre el uso y la actividad del almacenamiento de objetos. Para obtener más información, consulte Evaluación de la actividad y el uso de almacenamiento con S3 Storage Lens. Para obtener una lista completa de las métricas, consulte el Glosario de métricas de Lente de almacenamiento de S3.

El propietario del bucket, la Cuenta de AWS que creó el bucket (cuenta raíz) y todos los usuarios autorizados pueden habilitar el control de versiones. Sin embargo, solo el propietario del bucket (cuenta raíz) puede habilitar la eliminación de MFA. Para obtener más información, consulte Protección del acceso a AWS utilizando MFA en el blog de seguridad de AWS.

> **Note**
>
> Para utilizar la eliminación con MFA con el control de versiones, habilite `MFA Delete`. Sin embargo, no puede habilitar `MFA Delete` mediante la AWS Management Console. Para ello debe utilizar AWS Command Line Interface (AWS CLI) o la API.
>
> Para obtener ejemplos de cómo utilizar la eliminación con MFA con el control de versiones, consulte la sección de ejemplos en el tema Habilitar el control de versiones en buckets.
>
> No se puede utilizar la eliminación de MFA con configuraciones del ciclo de vida. Para obtener más información sobre las configuraciones del ciclo de vida y cómo interactúan con otras configuraciones, consulte Configuraciones del ciclo de vida y otras configuraciones del bucket.

Para habilitar o deshabilitar la eliminación de MFA, utilice la misma API que utiliza para configurar el control de versiones en un bucket. Amazon S3 almacena la configuración de la eliminación de MFA en el mismo subrecurso de control de versiones que almacena el estado del control de versiones del bucket.

```
<VersioningConfiguration xmlns="http://s3.amazonaws.com/doc/2006-03-01/">
  <Status>VersioningState</Status>
```

```
  <MfaDelete>MfaDeleteState</MfaDelete>
</VersioningConfiguration>
```

Para usar la eliminación de MFA, puede usar hardware o un dispositivo virtual de MFA para generar un código de autenticación. En el siguiente ejemplo se muestra un código de autenticación generado en un dispositivo de hardware.

La eliminación de MFA y el acceso a la API protegido con MFA son características cuyo objetivo es proporcionar protección en distintos escenarios. Puede configurar la eliminación de MFA en un bucket para garantizar que los datos en el bucket no se puedan eliminar por accidente. El acceso a la API protegido por MFA se usa para aplicar otro factor de autenticación (código de MFA) al obtener acceso a recursos de Amazon S3 confidenciales. Se puede requerir que todas las operaciones relacionadas con estos recursos de Amazon S3 se realicen utilizando credenciales temporales creadas con MFA. Para ver un ejemplo, consulte Exigir MFA.

Para obtener más información acerca de cómo comprar y activar un dispositivo de autenticación, consulte Multi-factor authentication.

Para habilitar el control de versiones de S3 y configurar la eliminación de MFA

Uso de la AWS CLI

En el siguiente ejemplo se habilita el control de versiones de S3 y la eliminación de autenticación multifactor (MFA) en un bucket.

```
aws s3api put-bucket-versioning --bucket amzn-s3-demo-bucket1 --versioning-
configuration Status=Enabled,MFADelete=Enabled --mfa "SERIAL 123456"
```

Uso de la API de REST

Para obtener más información sobre la especificación de la eliminación de MFA mediante la API de REST de Amazon S3, consulte la referencia de la API PutBucketVersioning de Amazon Simple Storage Service.

Trabajar con objetos en un bucket con control de versiones habilitado

Los objetos que se almacenan en un bucket de Amazon S3 antes de establecer el estado del control de versiones tienen un ID de versión de `null`. Al habilitar el control de versiones, los objetos existentes en el bucket no cambian. Lo que cambia es la forma en la que Amazon S3 administrará los objetos en las solicitudes futuras.

Transición de versiones de objetos

Puede definir reglas de configuración de ciclo de vida para objetos que tengan un ciclo de vida bien definido para realizar la transición de versiones de un objeto a la clase de almacenamiento S3 Glacier Flexible Retrieval en un momento específico del periodo de vida del objeto. Para obtener más información, consulte Administración del ciclo de vida del almacenamiento.

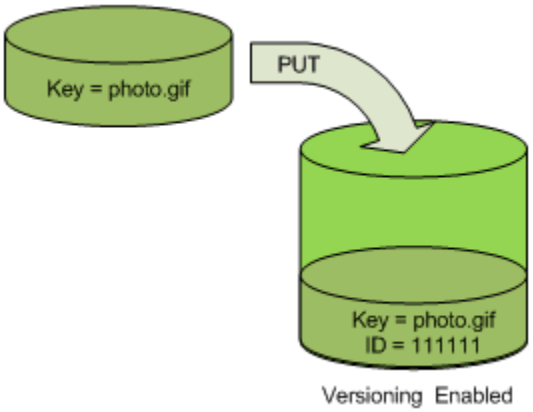
Los temas de esta sección explican varias operaciones con objetos en un bucket con control de versiones habilitado. Para obtener más información sobre el control de versiones, consulte Usar el control de versiones en buckets de S3.

Temas

Agregar objetos a buckets con control de versiones habilitado

Al habilitar el control de versiones en un bucket, Amazon S3 agrega automáticamente un ID de versión exclusivo a todos los objetos almacenados (con `PUT`, `POST` o `CopyObject`) en el bucket.

En el siguiente gráfico se muestra que Amazon S3 agrega un ID de versión exclusivo a un objeto cuando se agrega a un bucket con control de versiones activado.



> **Note**
>
> Los valores del ID de versión que asigna Amazon S3 son seguros para URL (se pueden usar como parte de un URI).

Para obtener más información sobre el control de versiones, consulte Usar el control de versiones en buckets de S3. Puede agregar versiones de objetos a un bucket habilitado para el control de versiones mediante la consola, los SDK de AWS y la API de REST.

Uso de la consola de

Para obtener instrucciones, consulte Carga de objetos.

Uso de los AWS SDK

Para ver ejemplos de cómo cargar objetos con los SDK de AWS para Java, .NET y PHP, consulte Carga de objetos. Los ejemplos para cargar objetos en buckets sin control de versiones y con control de versiones activado son iguales, aunque en el caso de los buckets con control de versiones activado, Amazon S3 asigna un número de versión. De lo contrario, el número de versión es un valor nulo.

Para obtener información acerca del uso de otros SDK de AWS, consulte el Centro de desarrolladores de AWS.

Uso de la API de REST

Agregar objetos a buckets con control de versiones habilitado

1. Habilite el control de versiones en un bucket con una solicitud `PutBucketVersioning`.

Para obtener más información, consulte PutBucketVersioning en la Referencia de la API de Amazon Simple Storage Service.

2. Envíe una solicitud `PUT`, `POST` o `CopyObject` para almacenar un objeto en el bucket.

Al agregar un objeto a un bucket con control de versiones activado, Amazon S3 devuelve el ID de versión del objeto en el `x-amz-version-id` encabezado de respuesta, por ejemplo:

```
x-amz-version-id: 3/L4kqtJlcpXroDTDmJ+rmSpXd3dIbrHY
```

Mostrar objetos en un bucket con control de versiones habilitado

En esta sección se presenta un ejemplo de cómo listar versiones de un objeto en un bucket con control de versiones habilitado. Amazon S3 almacena la información de versión de un objeto en el subrecurso de versiones asociado con el bucket. Para obtener más información, consulte Opciones de configuración de buckets. Para enumerar los objetos de un bucket con control de versiones activado, necesita el permiso `ListBucketVersions`.

Uso de la consola de S3

Siga estos pasos para utilizar la consola de Amazon S3 a fin de ver las diferentes versiones de un objeto.

Para ver múltiples versiones de un objeto

1. Inicie sesión en la AWS Management Console y abra la consola de Amazon S3 en https://console.aws.amazon.com/s3/.
2. En la lista Buckets (Buckets), elija el nombre del bucket que contiene el objeto.
3. Para ver una lista de las versiones de los objetos en el bucket, elija el modificador Show versions (Mostrar versiones).

   Para cada versión de objeto, la consola muestra un ID de versión único, la fecha y la hora en que se creó la versión del objeto, y otras propiedades. (Los objetos almacenados en un bucket antes de establecer el estado del control de versiones tienen el ID de versión null (nulo)).

   Para listar los objetos sin las versiones, elija el modificador List versions (Enumerar versiones) .

También puede ver, descargar y eliminar las versiones de los objetos en el panel de información general de objetos de la consola. Para obtener más información, consulte Visualización de información general sobre objetos en la consola de Amazon S3.

> **Note**
>
> Para acceder a versiones de objetos anteriores a 300 versiones, debe usar la AWS CLI o la URL del objeto.

> **Important**
>
> Solo puede anular la eliminación de un objeto si se ha eliminado en su última versión (la más reciente). No puede anular la eliminación de una versión anterior de un objeto que se haya eliminado. Para obtener más información, consulte Usar el control de versiones en buckets de S3.

Uso de los AWS SDK

Los ejemplos en esta sección muestran cómo recuperar un listado de objetos de un bucket con control de versiones habilitado. Cada solicitud devuelve hasta 1000 versiones, a menos que especifique un número más bajo. Si el bucket tiene más versiones que ese límite, tendrá que enviar varias solicitudes para recuperar la lista de todas las versiones. Este proceso de devolución de resultados en "páginas" se llama paginación.

Para mostrar cómo funciona la paginación, los ejemplos limitan cada respuesta a dos versiones de objetos. Después de recuperar la primera página de resultados, cada ejemplo realiza una comprobación para determinar si se truncó la lista de la versión. Si fue así, el ejemplo continúa recuperando páginas hasta que se hayan recuperado todas las versiones.

> **Note**
>
> El siguiente ejemplo también funciona con un bucket que no tiene habilitado el control de versiones u objetos que no tienen versiones individuales. En esos casos, Amazon S3 devuelve un listado de objetos con un ID de versión `null`.

Para obtener información acerca del uso de otros SDK de AWS, consulte el Centro de desarrolladores de AWS.

Java

Para obtener instrucciones sobre cómo crear y probar una muestra funcional, consulte Introducción en la Guía del desarrollador de AWS SDK for Java.

```
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

Para obtener información acerca de cómo configurar y ejecutar ejemplos de código, consulte Introducción al SDK de AWS para .NET en la Guía para desarrolladores del SDK de AWS para .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);
            }
        }
    }
}
```

Uso de la API de REST

Example — Listar todas las versiones de objetos en un bucket

Para enumerar todas las versiones de todos los objetos de un bucket, use el subrecurso `versions` en una solicitud `GET Bucket`. Amazon S3 solo puede recuperar un máximo de 1000 objetos, y cada versión del objeto cuenta como un objeto. Por tanto, si un bucket contiene dos claves (por ejemplo, `photo.gif` y `picture.jpg`), la primera clave tiene 990 versiones y la segunda tiene 400 versiones, una única solicitud recuperaría las 990 versiones de `photo.gif` y solo las 10 versiones más recientes de `picture.jpg`.

Amazon S3 devuelve las versiones de objetos en el orden en el que se almacenaron y devuelve las almacenadas más recientemente primero.

En una solicitud `GET Bucket`, incluya el subrecurso `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 — Recuperar todas las versiones de una clave

Para recuperar un subconjunto de las versiones de un objeto, use los parámetros de la solicitud `GET Bucket` Bucket. Para obtener más información, consulte GET Bucket.

1. Establezca el `prefix` parámetro en la clave del objeto que quiera recuperar.
2. Envíe una solicitud `GET Bucket` con el subrecurso `versions` y `prefix`.

   ```
   GET /?versions&prefix=objectName HTTP/1.1
   ```

Example — Recuperar objetos mediante un prefijo

En el siguiente ejemplo se recuperan objetos cuya clave es o comienza por `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=
```

Puede usar los otros parámetros de solicitud para recuperar un subconjunto de todas las versiones del objeto. Para obtener más información, consulte GET Bucket en la Referencia de la API de Amazon Simple Storage Service.

Example — Recuperar un listado de objetos adicionales si la respuesta está truncada

Si el número de objetos que se pueden devolver con una solicitud `GET` supera el valor de `max-keys`, la respuesta contendrá `<isTruncated>true</isTruncated>` e incluirá la primera clave (en `NextKeyMarker`) y el primer ID de versión (en `NextVersionIdMarker`) que se ajusten a la solicitud, pero no se hayan devuelto. Puede usar esos valores como posición de inicio en una solicitud subsiguiente para recuperar los objetos adicionales que se ajusten a la solicitud `GET`.

Puede usar el siguiente proceso para recuperar objetos adicionales que se ajusten a la solicitud `GET Bucket versions` original desde un bucket. Para obtener más información sobre `key-marker`, `version-id-marker`, `NextKeyMarker` y `NextVersionIdMarker`, consulte GET Bucket en la Referencia de la API de Amazon Simple Storage Service.

Las siguientes son respuestas adicionales que satisfacen la solicitud `GET` original:

- Establezca el valor de `key-marker` de acuerdo con la clave devuelta en `NextKeyMarker` en la respuesta anterior.
- Establezca el valor de `version-id-marker` de acuerdo con el ID de versión devuelto en `NextVersionIdMarker` en la respuesta anterior.
- Envíe una solicitud `GET Bucket versions` con el subrecurso `key-marker` y `version-id-marker`.

Example — Recuperar objetos que comienzan con una clave y un ID de versión específicos

```
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=
```

Uso de la AWS CLI

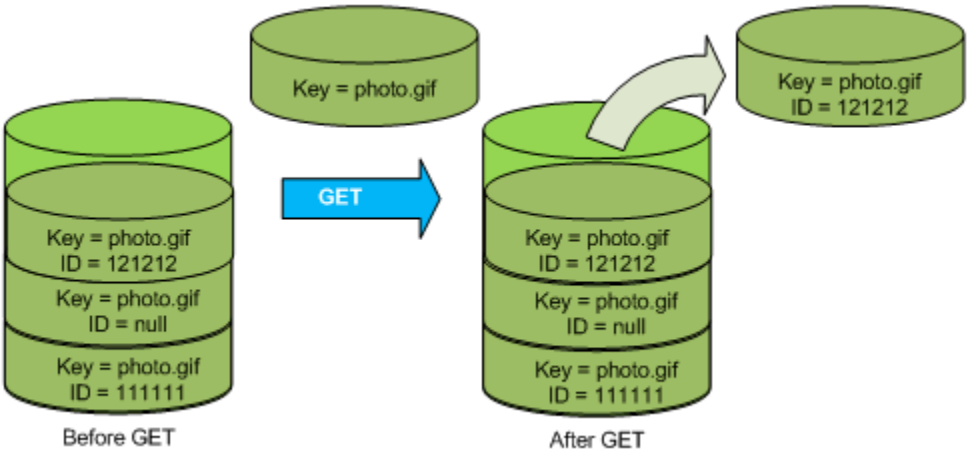
El siguiente comando devuelve los metadatos de todas las versiones de los objetos de un bucket.

```
aws s3api list-object-versions --bucket amzn-s3-demo-bucket1
```

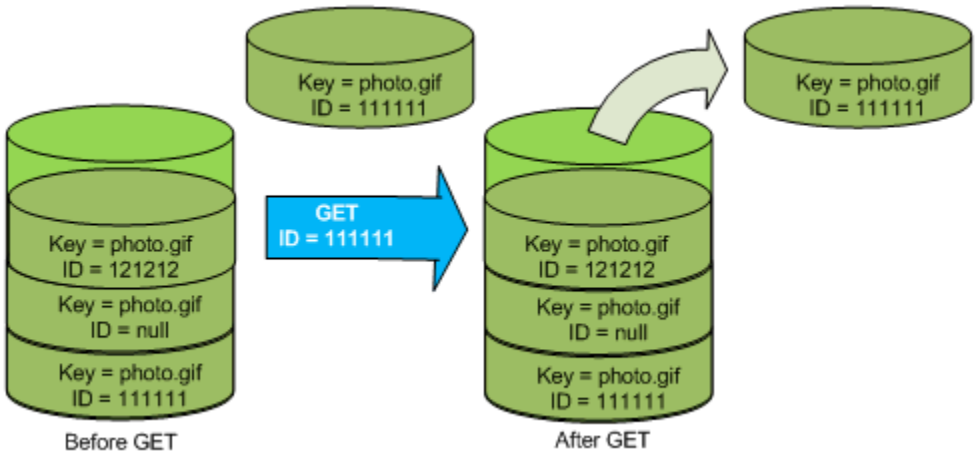
Para obtener más información sobre `list-object-versions`, consulte list-object-versions en la Referencia de los comandos de AWS CLI.

Recuperar versiones de objetos de un bucket habilitado para el control de versiones

El control de versiones en Amazon S3 es una forma de mantener varias variantes de un objeto en el mismo bucket. Una solicitud `GET` sencilla recupera la versión actual de un objeto. El siguiente gráfico muestra cómo `GET` devuelve la versión actual del objeto, `photo.gif`.



Para recuperar una versión específica, debe especificar su ID de versión. El siguiente gráfico muestra cómo una solicitud `GET versionId` devuelve la versión especificada del objeto (no necesariamente la actual).



Puede recuperar versiones de objetos en Amazon S3 mediante la consola, los SDK de AWS o la API de REST.

> **Note**
>
> Para acceder a versiones de objetos anteriores a 300 versiones, debe usar la CLI de AWS o la URL del objeto.

Uso de la consola de S3

1. Inicie sesión en la AWS Management Console y abra la consola de Amazon S3 en https://console.aws.amazon.com/s3/.
2. En la lista Buckets (Buckets), elija el nombre del bucket que contiene el objeto.
3. En la lista Objects (Objetos), elija el nombre del objeto.
4. Elija Versions (Versiones).

   Amazon S3 muestra todas las versiones del objeto.
5. Active la casilla de verificación situada junto al ID de versión de las versiones que desea recuperar.
6. Elija (Actions) Acciones, elija Download (Descargar) y guarde el objeto.

También puede ver, descargar y eliminar las versiones de los objetos en el panel de información general de objetos. Para obtener más información, consulte Visualización de información general sobre objetos en la consola de Amazon S3.

> **Important**
>
> Solo puede anular la eliminación de un objeto si se ha eliminado en su última versión (la más reciente). No puede anular la eliminación de una versión anterior de un objeto que se haya eliminado. Para obtener más información, consulte Usar el control de versiones en buckets de S3.

Uso de los AWS SDK

Los ejemplos para cargar objetos en buckets sin control de versiones y habilitados para el control de versiones son los mismos. Sin embargo, para los buckets habilitados para el control de versiones, Amazon S3 asigna un número de versión. De lo contrario, el número de versión es un valor nulo.

Para obtener ejemplos de la descarga de objetos mediante los SDK de AWS para Java, .NET y PHP, consulte Descarga de objetos.

Para ver ejemplos de cómo mostrar la versión de los objetos mediante los SDK de AWS para .NET y Rust, consulte Listar la versión de los objetos en un bucket de Amazon S3.

Uso de la API de REST

Para recuperar una versión de objeto específica:

1. Establezca el parámetro `versionId` según el ID de la versión del objeto que quiera recuperar.
2. Envíe una solicitud `GET Object versionId`.

Example — Recuperar un objeto con control de versiones

La siguiente solicitud recupera la versión `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=
```

Solo se pueden recuperar los metadatos de un objeto (no el contenido). Para obtener información, consulte the section called “Recuperando metadatos de versión”.

Para obtener información sobre cómo restaurar una versión de objeto anterior, consulte the section called “Restaurar versiones anteriores”.

Recuperar los metadatos de una versión de un objeto

Si solo quiere recuperar los metadatos de un objeto (y no su contenido), puede usar la operación `HEAD`. De forma predeterminada, obtendrá los metadatos de la versión más reciente. Para recuperar los metadatos de una versión de objeto específica, debe especificar su ID de versión.

Para recuperar los metadatos de una versión de objeto:

1. Establezca el parámetro `versionId` según el ID de la versión del objeto cuyos metadatos quiera recuperar.
2. Envíe una solicitud `HEAD Object versionId`.

Example — Recuperar los metadatos de un objeto con control de versiones

La siguiente solicitud recupera los metadatos de la versión `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=
```

A continuación se muestra una respuesta de ejemplo.

```
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
```

Restaurar versiones anteriores

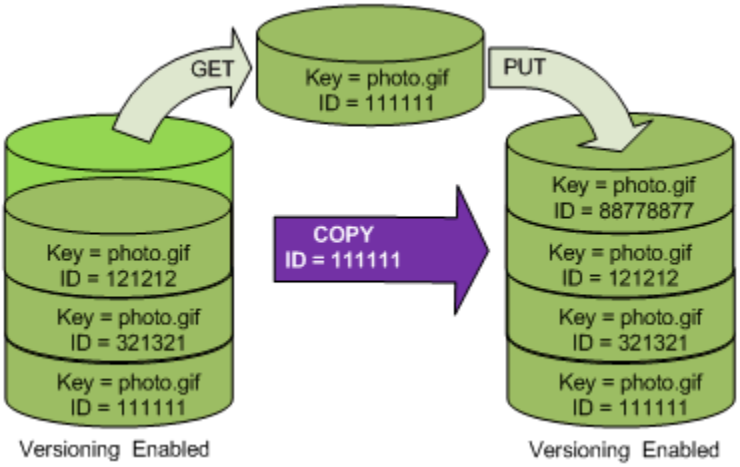
Se puede utilizar el control de versiones para recuperar versiones anteriores de un objeto. Existen dos enfoques para hacerlo:

- Copie una versión anterior del objeto en el mismo bucket.

  El objeto copiado se convierte en la versión actual del mismo, y se conservan todas las versiones del objeto.
- Elimine permanentemente la versión actual del objeto.

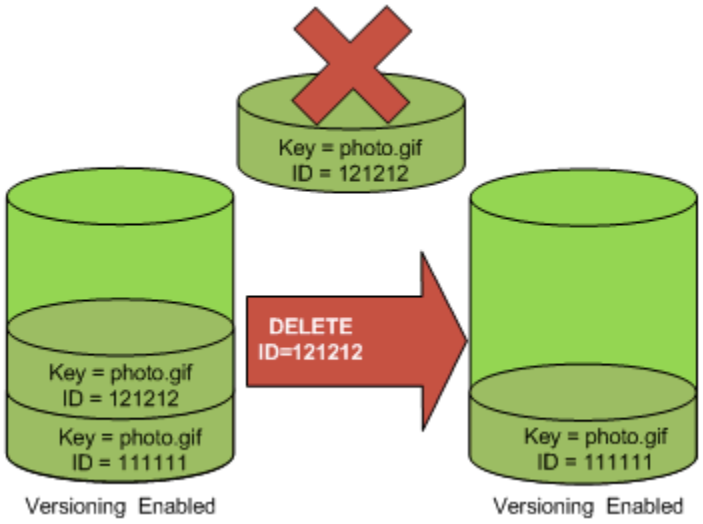
  Al eliminar la versión actual del objeto, en efecto, estará convirtiendo la versión anterior en la versión actual del mismo.

Dado que se conservan todas las versiones de los objetos, puede hacer que cualquier versión sea la versión actual copiando una versión específica del objeto en el mismo bucket. En el siguiente gráfico, el objeto de origen (ID=111111) se copia en el mismo bucket. Amazon S3 facilita un nuevo ID (88778877) y se convierte en la versión actual del objeto. Por tanto, el bucket contendrá tanto la versión original del objeto (111111) como su copia (88778877). Para obtener más información acerca de cómo obtener una versión anterior y, a continuación, cargarla para convertirla en la versión actual, consulte Recuperación de versiones de objetos de un bucket habilitado para el control de versiones y Cargar objetos.



Un subsiguiente GET recupera la versión 88778877.

En el siguiente gráfico se muestra cómo eliminar la versión actual (121212) de un objeto, lo que deja la versión anterior (111111) como objeto actual. Para obtener más información acerca de cómo eliminar un objeto, consulte Supresión de un solo objeto.



Un subsiguiente GET recupera la versión 111111.

> **Note**
>
> Para restaurar las versiones de los objetos en lotes, puede utilizar la operación `CopyObject`. Mediante la operación `CopyObject`, se copia cada objeto especificado en el manifiesto. Sin embargo, tenga en cuenta que los objetos no se copian necesariamente en el mismo orden en el que aparecen en el manifiesto. Para los buckets con versiones, si es importante conservar el orden de versiones actual/no actual, primero debe copiar todas las versiones no actuales. Luego, una vez finalizado el primer trabajo, copie las versiones actuales en un trabajo posterior.

Restaurar versiones anteriores de objetos

Uso de la consola de S3

1. Inicie sesión en la AWS Management Console y abra la consola de Amazon S3 en https://console.aws.amazon.com/s3/.
2. En la lista Buckets (Buckets), elija el nombre del bucket que contiene el objeto.
3. En la lista Objects (Objetos), elija el nombre del objeto.
4. Elija Versions (Versiones).

   Amazon S3 muestra todas las versiones del objeto.

5. Active la casilla de verificación situada junto al ID de versión de las versiones que desea recuperar.
6. Elija (Actions) Acciones, elija Download (Descargar) y guarde el objeto.

También puede ver, descargar y eliminar las versiones de los objetos en el panel de información general de objetos. Para obtener más información, consulte Visualización de información general sobre objetos en la consola de Amazon S3.

> **Important**
>
> Solo puede anular la eliminación de un objeto si se ha eliminado en su última versión (la más reciente). No puede anular la eliminación de una versión anterior de un objeto que se haya eliminado. Para obtener más información, consulte Usar el control de versiones en buckets de S3.

Uso de los AWS SDK

Para obtener información acerca del uso de otros SDK de AWS, consulte el Centro de desarrolladores de AWS.

Python

En el siguiente ejemplo de código de Python, se restaura la versión anterior de un objeto versionado; para ello, se eliminan todas las versiones que se produjeron después de la versión de restauración especificada.

```
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}."
        )
```

Eliminar versiones de objetos de un bucket con control de versiones habilitado

Puede eliminar versiones de objetos de los buckets de Amazon S3 siempre que lo desee. Además, también puede definir normas de configuración de ciclo de vida para objetos que tengan un ciclo de vida bien definido, de modo que solicite a Amazon S3 expirar versiones de objetos actuales o eliminar permanentemente versiones de objetos no actuales. Cuando en su bucket está habilitado o suspendido el control de versiones, las acciones de configuración de ciclo de vida funcionan así:

- La `Expiration` acción se aplica a la versión actual del objeto. En lugar de eliminar la versión actual, Amazon S3 conserva la versión actual como versión no actual agregándole un marcador de eliminación, con lo cual se convierte en la versión actual.
- La acción `NoncurrentVersionExpiration` se aplica a las versiones no actuales del objeto, y Amazon S3 elimina permanentemente estas versiones de objeto. No puede recuperar objetos eliminados permanentemente.

Para obtener más información acerca del ciclo de vida de S3, consulte Administración del ciclo de vida del almacenamiento y Ejemplos de configuración de S3 Lifecycle.

Para ver cuántas versiones de objetos actuales y no actuales tienen los buckets, puede usar las métricas de Lente de almacenamiento de Amazon S3. Lente de almacenamiento de S3 es una función de análisis de almacenamiento en la nube que puede utilizar para obtener visibilidad en toda la organización sobre el uso y la actividad del almacenamiento de objetos. Para obtener más información, consulte Uso de Lente de almacenamiento de S3 para optimizar los costos de almacenamiento. Para obtener una lista completa de las métricas, consulte el Glosario de métricas de Lente de almacenamiento de S3.

> **Note**
>
> Se aplican tasas normales de Amazon S3 por cada versión de un objeto almacenado y transferido, incluidas las versiones de objetos no actuales. Para obtener más información, consulte Precios de Amazon S3.

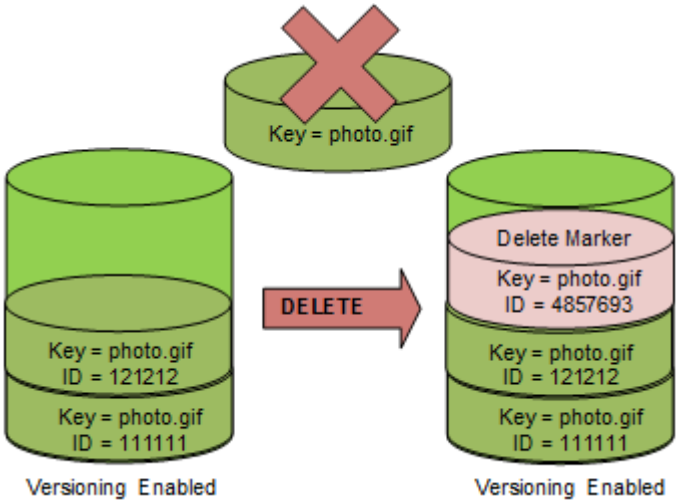
Eliminar casos de uso de solicitudes

Una solicitud `DELETE` presenta los siguientes casos de uso:

- Cuando está habilitado el control de versiones, un `DELETE` simple no puede eliminar permanentemente un objeto. (Una solicitud `DELETE` simple es una solicitud que no especifica un ID de versión). En su lugar, Amazon S3 inserta un marcador de eliminación en el bucket, y ese marcador se convierte en la versión actual del objeto con un nuevo ID.

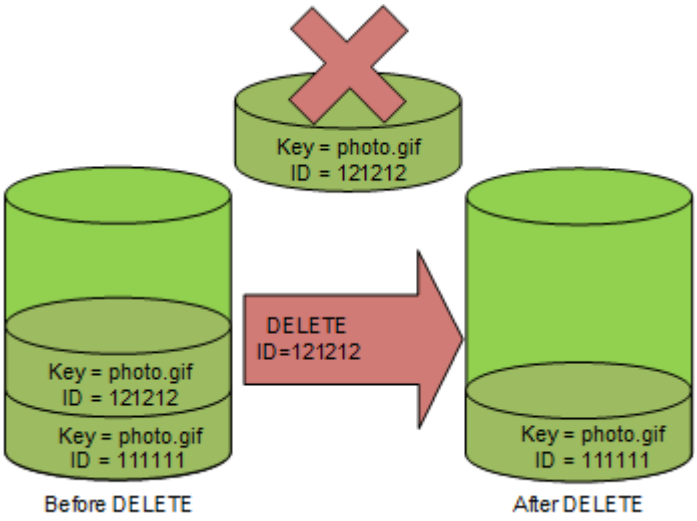
  Cuando intenta `GET` un objeto cuya versión actual es un marcador de eliminación, Amazon S3 se comporta como si el objeto se hubiera eliminado (aunque no sea el caso) y devuelve un error 404. Para obtener más información, consulte Trabajar con marcadores de eliminación.

El siguiente gráfico muestra cómo una solicitud `DELETE` simple no elimina realmente el objeto especificado. En su lugar, Amazon S3 inserta un marcador de eliminación.



- Para eliminar de forma permanente objetos con control de versiones, debe usar `DELETE Object versionId`.

  El siguiente gráfico muestra cómo la eliminación de una versión de objeto específica elimina permanentemente el objeto.



Para eliminar versiones de objetos

Puede eliminar versiones de objetos en Amazon S3 mediante la consola, los SDK de AWS, la API de REST o AWS Command Line Interface.

Uso de la consola de S3

1. Inicie sesión en la AWS Management Console y abra la consola de Amazon S3 en https://console.aws.amazon.com/s3/.
2. En la lista Buckets (Buckets), elija el nombre del bucket que contiene el objeto.
3. En la lista Objects (Objetos), elija el nombre del objeto.
4. Elija Versions (Versiones).

   Amazon S3 muestra todas las versiones del objeto.
5. Active la casilla de verificación situada junto al ID de versión de las versiones que desea eliminar de manera permanente.
6. Elija Eliminar.
7. En Permanently delete objects? (¿Eliminar objetos de forma permanente?) , ingrese **`permanently delete`**.

   > **Warning**
   >
   > Cuando elimina de forma permanente una versión de un objeto, la acción no se puede deshacer.
8. Elija Eliminar objetos.

   Amazon S3 elimina la versión del objeto.

Uso de los AWS SDK

Para ver ejemplos de cómo eliminar objetos con los AWS SDK para Java, .NET y PHP, consulte Eliminación de objetos de Amazon S3. Los ejemplos para eliminar objetos en buckets sin control de versiones y habilitados para la control de versiones son los mismos. Sin embargo, para los buckets habilitados para el control de versiones, Amazon S3 asigna un número de versión. De lo contrario, el número de versión es un valor nulo.

Para obtener información acerca del uso de otros SDK de AWS, consulte el Centro de desarrolladores de AWS.

Python

En el siguiente ejemplo de código de Python, se elimina permanente un objeto con control de versiones mediante la eliminación de todas sus versiones.

```
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
```

Uso de la API de REST

Para eliminar una versión específica de un objeto

- En un `DELETE`, especifique un ID de versión.

Example — Eliminar una versión específica

En el ejemplo siguiente se elimina la versión `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
```

Uso de la AWS CLI

El comando siguiente elimina un objeto denominado test.txt de un bucket denominado *amzn-s3-demo-bucket1*. Para quitar una versión específica de un objeto, debe ser el propietario del bucket y debe utilizar el subrecurso de del identificador de versión.

```
aws s3api delete-object --bucket amzn-s3-demo-bucket1 --key test.txt --version-
id versionID
```

Para obtener más información sobre `delete-object`, consulte delete-object en la Referencia de los comandos de AWS CLI.

Para obtener más información sobre cómo eliminar versiones de objetos, consulte los siguientes temas:

- Trabajar con marcadores de eliminación
- Borre marcadores de eliminación para convertir una versión anterior en una actual
- Eliminar un objeto de un bucket habilitado para la eliminación de MFA

Trabajar con marcadores de eliminación

Un marcador de eliminación en Amazon S3 es un marcador de posición (o marcador) para un objeto con control de versiones que se ha especificado en una solicitud `DELETE` simple. Una solicitud `DELETE` simple es una solicitud que no especifica un ID de versión. Dado que el objeto estaba en un bucket con control de versiones habilitado, el objeto no se elimina. Sin embargo, el marcador de eliminación hace que Amazon S3 se comporte como si el objeto se hubiese eliminado. Puede utilizar una llamada `DELETE` a la API de Amazon S3 en un marcador de eliminación. Para ello, deberá realizar la solicitud `DELETE` mediante un usuario o rol de AWS Identity and Access Management (IAM) que tenga los permisos apropiados.

Un marcador de eliminación tiene un nombre de clave (o clave) y un ID de versión al igual que cualquier otro objeto. Sin embargo, un marcador de eliminación se diferencia de otros objetos en los siguientes aspectos:

- Un marcador de eliminación no tiene datos asociados.
- Un marcador de eliminación no está asociado a un valor de lista de control de acceso (ACL).

- Si emite una solicitud `GET` para eliminar un marcador, la solicitud `GET` no recupera nada porque el marcador de eliminación no contiene datos. En concreto, si su solicitud `GET` no especifica un `versionId`, aparece un error 404 (no encontrado).

Los marcadores de eliminación acumulan un cargo mínimo por almacenamiento en Amazon S3. El tamaño de almacenamiento de un marcador de eliminación es igual al tamaño del nombre de clave del marcador de eliminación. Un nombre de clave es una secuencia de caracteres Unicode. La codificación UTF-8 del nombre de la clave añade entre 1 y 4 bytes de almacenamiento al bucket para cada carácter del nombre. Los marcadores de eliminación se almacenan en la clase de almacenamiento S3 Standard.

Si desea saber cuántos marcadores de eliminación tiene y en qué clase de almacenamiento están almacenados, puede usar la Lente de almacenamiento de Amazon S3. Para obtener más información, consulte Evaluación de la actividad y el uso de almacenamiento con Amazon S3 Storage Lens y Glosario de métricas de Amazon S3 Storage Lens.

Para obtener más información sobre nombres de clave, consulte Creación de nombres de clave de objeto. Para obtener información acerca de cómo eliminar marcadores de eliminación, consulte Gestión de marcadores de eliminación.

Solo Amazon S3 puede crear un marcador de eliminación, y lo hace cuando envía una solicitud `DeleteObject` a un objeto en un bucket con control de versiones habilitado o suspendido. El objeto especificado en la solicitud `DELETE` no se elimina realmente. Por el contrario, el marcador de eliminación se convierte en la versión actual del objeto. El nombre de la clave del objeto (o clave) se convierte en la clave del marcador de eliminación.

Cuando obtiene un objeto sin especificar un `versionId` en su solicitud, si su versión actual es un marcador de eliminación, Amazon S3 responde con lo siguiente:

- Un error 404 (No encontrado)
- Un encabezado de respuesta, `x-amz-delete-marker: true`

Cuando obtiene un objeto especificando un `versionId` en su solicitud, si la versión especificada es un marcador de eliminación, Amazon S3 responde con lo siguiente:

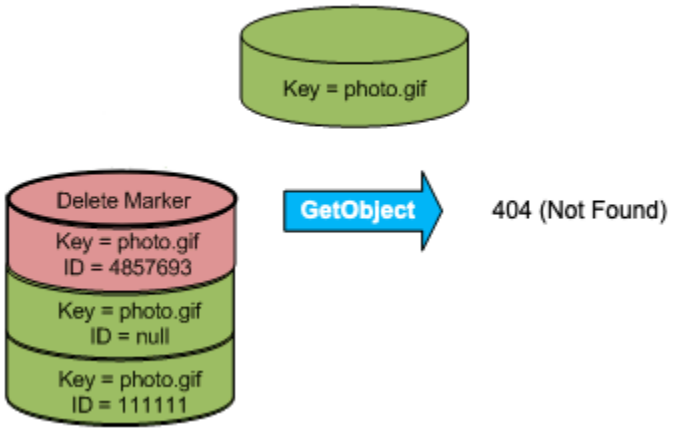
- Un error 405 (Método no permitido)
- Un encabezado de respuesta, `x-amz-delete-marker: true`

- Un encabezado de respuesta, `Last-Modified: timestamp` (solo cuando se utilizan las operaciones de la API HeadObject o GetObject)

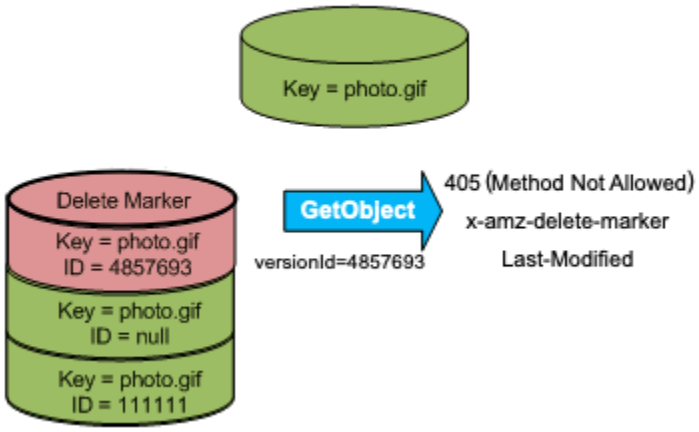
El encabezado de respuesta `x-amz-delete-marker: true` le indica que el objeto al que se ha obtenido acceso era un marcador de eliminación. Este encabezado de respuesta nunca devuelve `false`, porque cuando el valor es `false`, la versión actual o especificada del objeto no es un marcador de eliminación.

El encabezado de respuesta `Last-Modified` indica la hora de creación de los marcadores de eliminación.

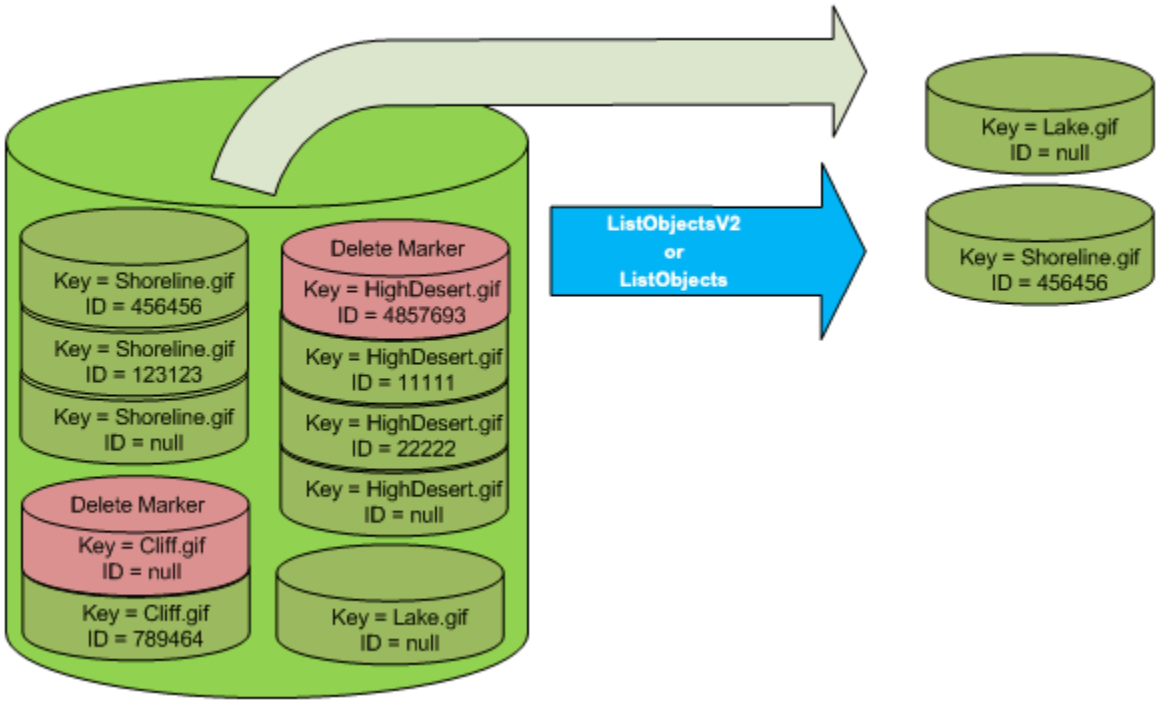
En el siguiente gráfico, se muestra cómo una llamada a la API `GetObject` en un objeto cuya versión actual es un marcador de eliminación responde con un error 404 (no encontrado) y en el encabezado de respuesta se incluye `x-amz-delete-marker: true`.



Si realiza una llamada `GetObject` a un objeto especificando un `versionId` en su solicitud y si la versión especificada es un marcador de eliminación, Amazon S3 responde con un error 405 (método no permitido) y los encabezados de respuesta incluyen `x-amz-delete-marker: true` y `Last-Modified: timestamp`.



Para enumerar todas las versiones de todos los objetos de un bucket, use el subrecurso `versions` en una solicitud ListObjectVersions. La siguiente figura muestra que una solicitud ListObjectsV2 o ListObjects no devuelve objetos cuya versión actual sea un marcador de eliminación.



Gestión de marcadores de eliminación

Configuración del ciclo de vida para limpiar automáticamente los marcadores de eliminación vencidos

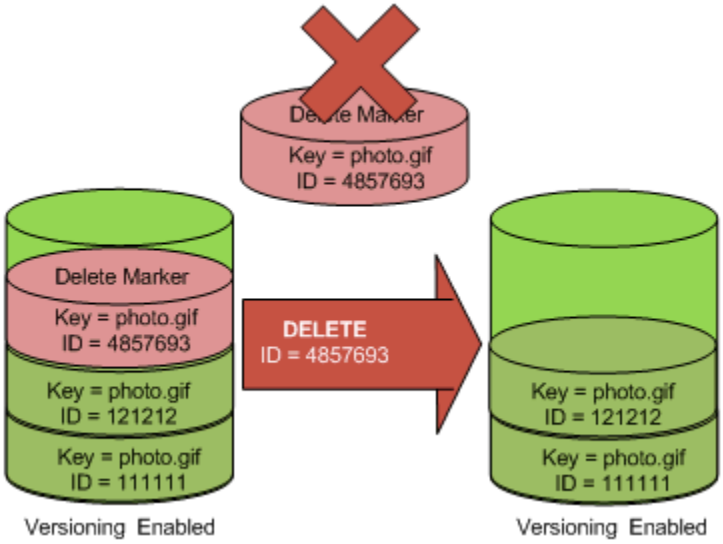
Un marcador de eliminación de objetos vencido es aquel en el que se eliminan todas las versiones de objetos y solo queda un único marcador de eliminación. Si la configuración de ciclo de vida está definida para eliminar las versiones actuales, o la acción `ExpiredObjectDeleteMarker` se ha

establecido explícitamente, Amazon S3 borra el marcador de eliminación del objeto vencido. Para ver un ejemplo, consulte Ejemplo 7: eliminar marcadores de eliminación de objetos que vencieron.

Borre marcadores de eliminación para convertir una versión anterior en una actual

Cuando se elimina un objeto en un bucket con control de versiones activado, todas las versiones permanecen en el bucket y Amazon S3 crea un marcador de eliminación para el objeto. Para anular la eliminación del objeto, debe eliminar este marcador de eliminación. Para obtener más información acerca del control de versiones y los marcadores de eliminación, consulte Usar el control de versiones en buckets de S3.

Para eliminar permanentemente un marcador de eliminación, se debe incluir el ID de versión en una solicitud `DeleteObject versionId`. En el siguiente gráfico se muestra cómo una solicitud `DeleteObject versionId` elimina permanentemente un marcador de eliminación.

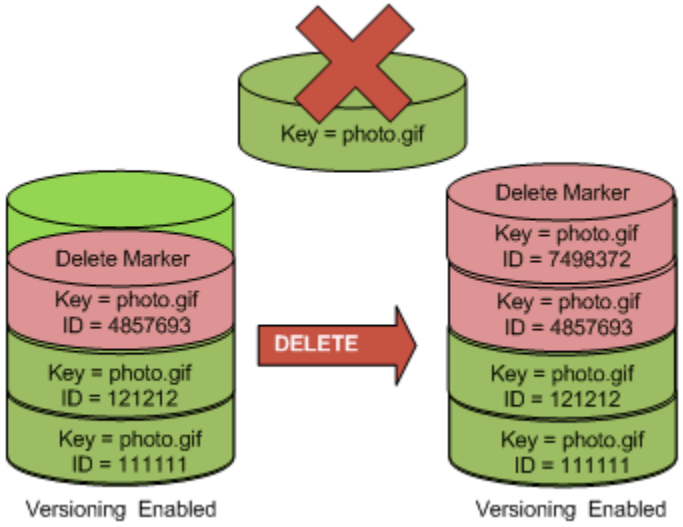


Al borrar el marcador de eliminación, ahora una solicitud `GET` simple devuelve el ID de versión actual (121212) del objeto.

> **Note**
>
> Si usa una solicitud `DeleteObject` para borrar un marcador de eliminación (sin especificar el ID de versión del marcador de eliminación), Amazon S3 no borra el marcador de eliminación, sino que `PUT`s (inserta) un nuevo marcador de eliminación.

Para borrar un marcador de eliminación con un ID de versión `NULL`, se debe pasar el `NULL` como el ID de versión en la solicitud `DeleteObject`. En la siguiente figura, se muestra cómo una solicitud

`DeleteObject` simple realizada sin un ID de versión donde la versión actual es un marcador de eliminación, no elimina nada, sino que agrega un marcador de eliminación adicional con un ID de versión único (7498372).



Uso de la consola de S3

Siga los pasos siguientes para recuperar objetos eliminados que no son carpetas del bucket de S3, incluidos los objetos que se encuentran dentro de esas carpetas.

1. Inicie sesión en la AWS Management Console y abra la consola de Amazon S3 en https://console.aws.amazon.com/s3/.
2. En la lista Buckets (Buckets), elija el nombre del bucket en cuestión.
3. Para ver una lista de las versiones de los objetos en el bucket, elija el modificador List versions (Listar versiones). Podrá ver los marcadores de eliminación de los objetos eliminados.
4. Para anular la eliminación de un objeto, debe eliminar su marcador de eliminación. Marque la casilla de verificación que aparece junto al delete marker (marcador de eliminación) del objeto que desee recuperar y, a continuación, elija Delete (Eliminar).
5. Confirme la eliminación en la página Delete objects (Eliminar objetos) .
    a. Para Permanently delete objects? (¿Eliminar objetos de forma permanente?), ingrese **`permanently delete`**.
    b. Elija Delete objects (Eliminar objetos).

> **Note**
>
> No puede usar la consola de Amazon S3 para anular la eliminación de carpetas. Debe utilizar la AWS CLI o el SDK. Para ver ejemplos, consulte ¿Cómo puedo recuperar un objeto de Amazon S3 que se eliminó en un bucket con control de versiones habilitado? en el Centro de conocimientos de AWS.

Uso de la API de REST

Para eliminar permanentemente un marcador de eliminación:

1. Establezca el parámetro `versionId` según el ID de la versión del marcador de eliminación que quiera eliminar.
2. Envíe una solicitud `DELETE Object versionId`.

Example — Eliminar un marcador de eliminación

En el siguiente ejemplo se elimina el marcador de eliminación para la versión 4857693 de `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=
```

Al eliminar un marcador de eliminación, Amazon S3 incluye en la respuesta:

```
204 NoContent
x-amz-version-id: versionID
x-amz-delete-marker: true
```

Uso de los AWS SDK

Para obtener información acerca del uso de otros SDK de AWS, consulte el Centro de desarrolladores de AWS.

Python

En el siguiente ejemplo de código de Python, se muestra cómo quitar un marcador de eliminación de un objeto y, por lo tanto, hacer que la versión no actual más reciente sea la versión actual del objeto.

```
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)
```

Eliminar un objeto de un bucket habilitado para la eliminación de MFA

Si la configuración del control de versiones de un bucket tiene la eliminación de MFA habilitada, el propietario del bucket debe incluir el encabezado de solicitud `x-amz-mfa` en las solicitudes para eliminar de forma permanente una versión de objeto o cambiar el estado del control de versiones del bucket. Las solicitudes que incluyen `x-amz-mfa` deben usar HTTPS.

El valor del encabezado es la concatenación del número de serie de su dispositivo de autenticación, un espacio y el código de autenticación que se muestra en él. Si no incluye este encabezado de solicitud, la solicitud producirá un error.

Para obtener más información acerca de los dispositivos de autenticación, consulte Multi-factor Authentication (Autenticación multifactor).

Example — Eliminar un objeto de un bucket habilitado para la eliminación de MFA

El siguiente ejemplo elimina `my-image.jpg` (con la versión especificada), que se encuentra en un bucket configurado con la eliminación con MFA habilitada.

Tenga en cuenta el espacio entre *`[SerialNumber]`* y *`[AuthenticationCode]`*. Para obtener más información, consulte DeleteObject en la Referencia de la API de 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=
```

Para obtener más información acerca de la habilitación de la eliminación con MFA, consulte Configurar la eliminación de MFA.

Configurar permisos de objeto con control de versiones

Los permisos para los objetos de Amazon S3 se establecen en el nivel de versión. Cada versión tiene su propio propietario de objeto. La Cuenta de AWS que crea la versión del objeto es la propietaria. Así, puede establecer distintos permisos para diferentes versiones del mismo objeto. Para hacerlo, debe especificar el ID de la versión del objeto cuyos permisos quiera establecer en una solicitud `PUT Object versionId acl`. Para obtener una descripción detallada e instrucciones sobre cómo usar las ACL, consulte Administración de identidades y accesos para Amazon S3.

Example — Establecer permisos para una versión de objeto

La siguiente solicitud establece los permisos del beneficiario, `BucketOwner@amazon.com`, como `FULL_CONTROL` en la clave, `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>
```

Del mismo modo, para obtener permisos para una versión de objeto específica, debe facilitar su ID de versión en una solicitud `GET Object versionId acl`. Debe incluir el ID de versión porque, de forma predeterminada, `GET Object acl` devuelve los permisos de la versión actual del objeto.

Example — Recuperar los permisos para una versión de objeto especificada

En el siguiente ejemplo, Amazon S3 devuelve los permisos para la clave, `my-image.jpg`, ID de versión, `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
```

Para obtener más información, consulte GetObjectAcl en la Referencia de la API de Amazon Simple Storage Service.

Trabajar con objetos en un bucket con control de versiones suspendido

En Amazon S3, puede suspender el control de versiones para evitar que se acumulen nuevas versiones del mismo objeto en un bucket. Puede hacer esto porque solo desea una sola versión de un objeto en un bucket. O bien, es posible que no desee acumular cargos para varias versiones.

Al suspender el control de versiones, los objetos existentes en el bucket no cambian. Lo que cambia es la forma en la que Amazon S3 administrará los objetos en las solicitudes futuras. Los temas de esta sección explican varias operaciones de objetos en un bucket con control de versiones suspendido, incluida la adición, recuperación y eliminación de objetos.

Para obtener más información sobre el control de versiones de S3, consulte Usar el control de versiones en buckets de S3. Para obtener más información sobre la recuperación de versiones de objetos, consulte Recuperar versiones de objetos de un bucket habilitado para el control de versiones.

Temas

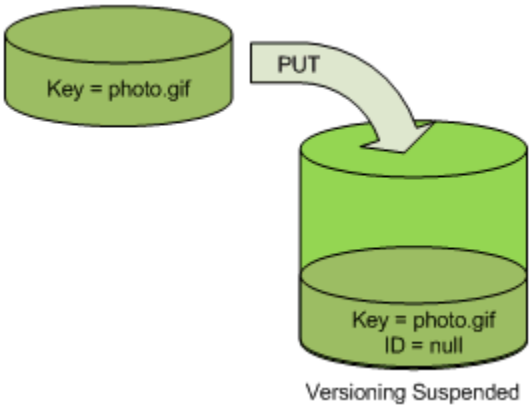
- Agregar objetos a buckets con control de versiones suspendido
- Recuperar objetos desde buckets con control de versiones suspendido
- Eliminar objetos de buckets con control de versiones suspendido

Agregar objetos a buckets con control de versiones suspendido

En Amazon S3 se pueden agregar objetos a buckets con control de versiones suspendido para crear el objeto con un ID de versión nulo o para sobrescribir las versiones del objeto con un ID de versión que coincida.

Al suspender el control de versiones en un bucket, Amazon S3 agrega automáticamente un `null` ID de versión a todos los objetos almacenados de forma subsecuente (con `PUT`, `POST` o `CopyObject`) en ese bucket.

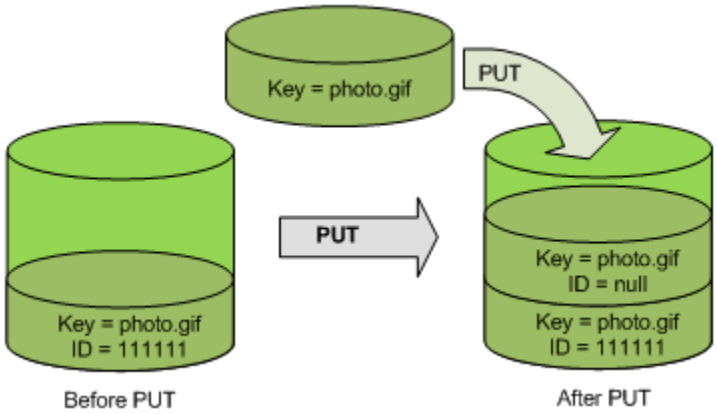
En el siguiente gráfico se muestra cómo Amazon S3 agrega un ID de versión `null` a un objeto cuando se agrega a un bucket con control de versiones suspendido.



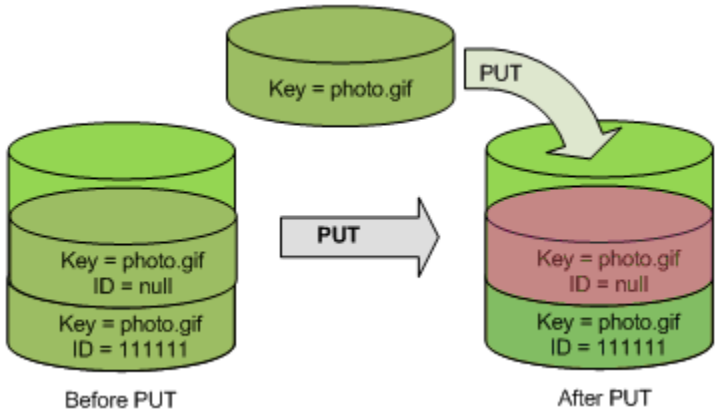
Si ya existe una versión nula en el bucket y agrega otro objeto con la misma clave, el objeto agregado sobrescribe la versión nula original.

Si hay objetos versionados en el bucket, la versión que someta a la operación `PUT` se convertirá en la versión actual del objeto. En el siguiente gráfico se muestra cómo agregar un objeto a un bucket que contiene objetos versionados no sobrescribe el objeto ya presente en el bucket.

En esta caso, la versión 111111 ya estaba en el bucket. Amazon S3 adjunta un ID de versión nulo al objeto que se agrega, y lo almacena en el bucket. La versión 111111 no se sobrescribe.



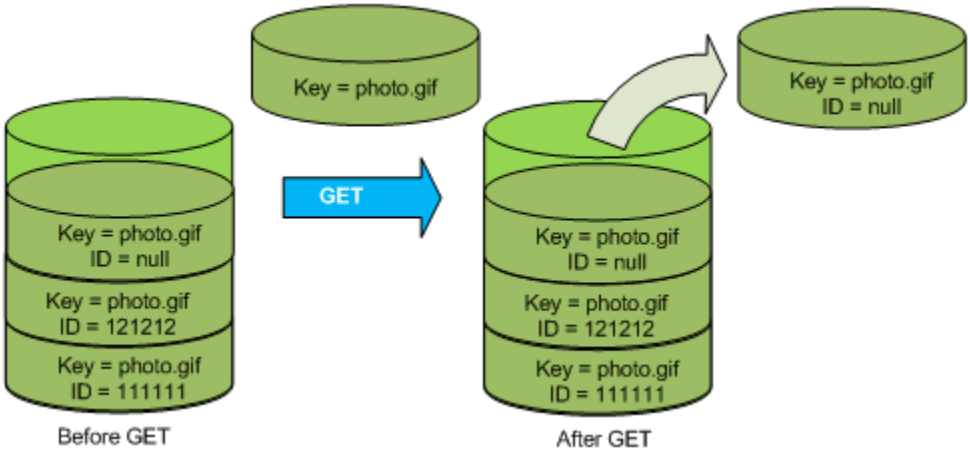
Si ya existe una versión nula en un bucket, la versión nula se sobrescribe, como se muestra en el siguiente gráfico.



Tenga en cuenta que aunque la clave y el ID de versión (`null`) de la versión nula son los mismos antes y después de la operación PUT, los contenidos de la versión nula almacenados en primer lugar en el bucket se sustituyen por los contenidos del objeto PUT en el bucket.

Recuperar objetos desde buckets con control de versiones suspendido

Una solicitud `GET Object` devuelve la versión actual de un objeto, haya activado el control de versiones en un bucket o no. El siguiente gráfico muestra cómo una solicitud `GET` sencilla devuelve la versión actual de un objeto.



Eliminar objetos de buckets con control de versiones suspendido

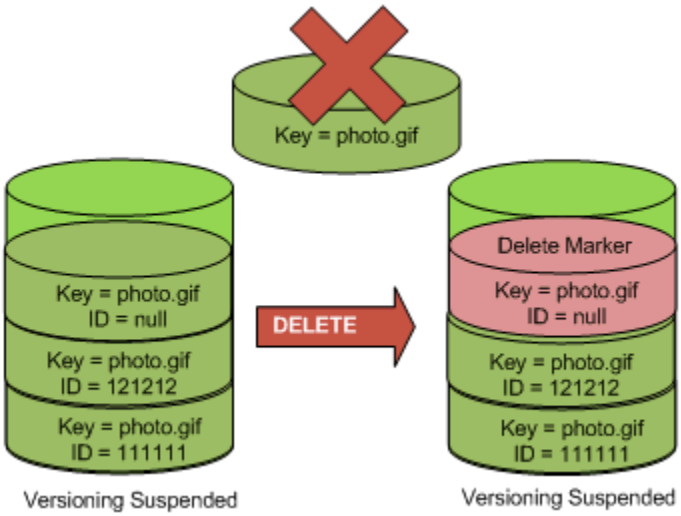
Puede eliminar objetos de buckets con control de versiones suspendido para eliminar un objeto con ID de versión nulo.

Si el control de versiones se suspende para un bucket, `DELETE` solicitará:

- Solo puede eliminar un objeto cuyo ID de versión sea `null`.
- No elimina ningún elemento si no hay una versión nula del objeto en el bucket.

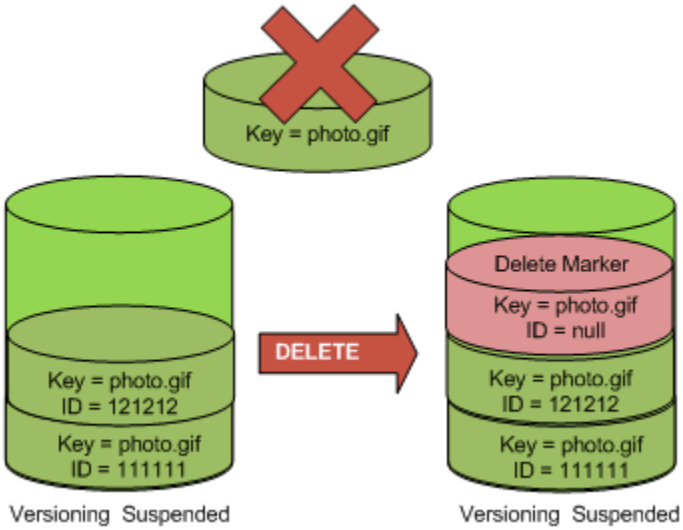
- Inserta un marcador de eliminación en el bucket.

La siguiente figura muestra cómo un simple `DELETE` elimina una versión nula. (Una solicitud `DELETE` simple es una solicitud que no especifica un ID de versión). Amazon S3 inserta un marcador de eliminación en su lugar con un ID de versión de `null`.



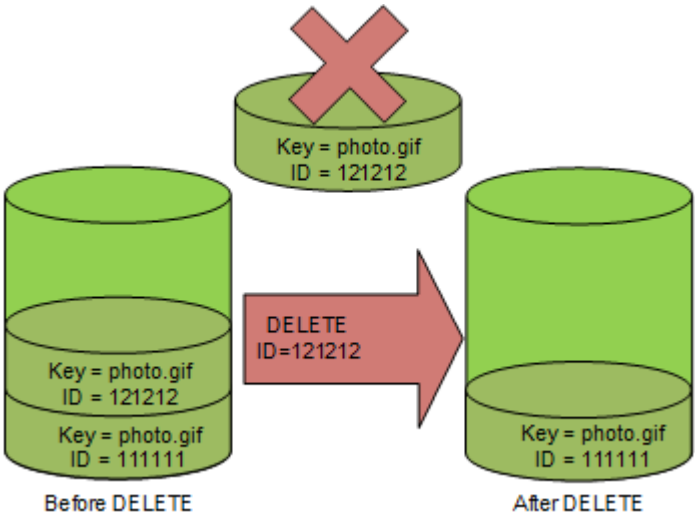
Recuerde que un marcador de eliminación no tiene ningún contenido, por lo que perderá el contenido de la versión nula cuando un marcador de eliminación la sustituya.

El siguiente gráfico muestra un bucket que no tiene una versión nula. En este caso, el `DELETE` no elimina nada, Amazon S3 inserta simplemente un marcador de eliminación.



Incluso en un bucket con control de versiones suspendido, el propietario del bucket puede eliminar permanentemente una versión específica; para ello, incluye el ID de versión en la solicitud `DELETE`.

En el siguiente gráfico, se muestra cómo la eliminación de una versión de objeto específica elimina el objeto de forma permanente. Solo el propietario de un bucket puede eliminar una versión de objeto específica.



Uso de AWS Backup para Amazon S3

Amazon S3 se integra de forma nativa con AWS Backup, un servicio completamente administrado basado en políticas que puede utilizar para definir de forma central políticas de copia de seguridad para proteger los datos de Amazon S3. Después de definir las políticas de copia de seguridad y asignar recursos de Amazon S3 a las políticas, AWS Backup automatiza la creación de copias de seguridad de Amazon S3 y almacena de forma segura las copias de seguridad en un almacén de copias de seguridad cifrado que haya designado en su plan de copia de seguridad.

Al utilizar AWS Backup para Amazon S3, puede realizar las siguientes acciones:

- Crear copias de seguridad continuas y copias de seguridad periódicas. Las copias de seguridad continuas son útiles para realizar una restauración en un punto del tiempo y las copias de seguridad periódicas son útiles para satisfacer sus necesidades de retención de datos a largo plazo.
- Automatizar la programación y la retención de copias de seguridad mediante la configuración centralizada de las políticas de copia de seguridad.
- Restaurar copias de seguridad de los datos de Amazon S3 en el punto del tiempo que especifique.

Junto con AWS Backup, puede utilizar el control de versiones de S3 y la replicación de S3 para ayudar a recuperarse de eliminaciones accidentales y realizar sus propias operaciones de recuperación automática.

Requisitos previos

Debe activar el control de versiones de S3 en el bucket para que AWS Backup pueda realizar una copia se seguridad.

> **Note**
>
> Recomendamos que establezca una regla de vencimiento del ciclo de vida para los buckets con control de versiones de los que se está realizando una copia de seguridad. Si no configura un periodo de vencimiento del ciclo de vida, los costos de almacenamiento de Amazon S3 podrían aumentar porque AWS Backup retendrá todas las versiones de los datos de Amazon S3.

Introducción

Para comenzar a trabajar con AWS Backup para Amazon S3, consulte Creación de copias de seguridad de Amazon S3 en la Guía para desarrolladores de AWS Backup.

Restricciones y limitaciones

Para conocer las limitaciones, consulte Creación de copias de seguridad de Amazon S3 en la Guía para desarrolladores de AWS Backup.

Trabajar con objetos archivados

Para reducir los costos de almacenamiento de los objetos a los que se accede con poca frecuencia, puede archivar dichos objetos. Cuando archiva un objeto, se traslada a un almacenamiento de bajo costo, lo que significa que no puede acceder a él en tiempo real.

Aunque los objetos archivados no son accesibles en tiempo real, puede restaurarlos en minutos u horas, según la clase de almacenamiento. Puede restaurar un objeto archivado mediante la consola de Amazon S3, Operaciones por lotes de S3, la API de REST, los SDK de AWS y la AWS Command Line Interface (AWS CLI). Para obtener instrucciones, consulte Restauración de un objeto archivado.

Los objetos de Amazon S3 de las siguientes clases o niveles de almacenamiento se archivan y no son accesibles en tiempo real:

- La clase de almacenamiento S3 Glacier Flexible Retrieval
- La clase de almacenamiento S3 Glacier Deep Archive
- La capa de acceso de archivo de S3 Intelligent-Tiering
- El nivel S3 Intelligent-Tiering Deep Archive Access

Para restaurar los objetos archivados, se debe hacer lo siguiente:

- Para los objetos de las clases de almacenamiento S3 Glacier Flexible Retrieval o S3 Glacier Deep Archive, se debe iniciar una solicitud de restauración y esperar hasta que esté disponible una copia temporal del objeto. Cuando se crea una copia temporal del objeto restaurado, la clase de almacenamiento del objeto sigue siendo la misma. (Una solicitud de operación de API HeadObject o GetObject devuelve S3 Glacier Flexible Retrieval o S3 Glacier Deep Archive como clase de almacenamiento).
- En el caso de los objetos de las capas de acceso de archivo o archivo profundo de S3 Intelligent-Tiering, se debe iniciar una solicitud de restauración y esperar hasta que el objeto se mueva a la capa de acceso frecuente.

Para obtener más información acerca de cómo se comparan todas las clases de almacenamiento de Amazon S3, consulte Uso de las clases de almacenamiento de Amazon S3. Para obtener más información acerca de S3 Intelligent-Tiering, consulte the section called “Cómo funciona S3 Intelligent-Tiering”.

Restauración de objetos desde S3 Glacier

Cuando utiliza S3 Glacier Flexible Retrieval o S3 Glacier Deep Archive, Amazon S3 restaura una copia temporal del objeto solo durante la duración especificada. Después, elimina la copia del objeto restaurada. Puede modificar el periodo de vencimiento de una copia restaurada mediante la reemisión de una solicitud de restauración. En este caso, Amazon S3 actualiza el periodo de vencimiento en relación con el momento actual.

> **Note**
>
> Cuando restaura un archivo archivado desde S3 Glacier Flexible Retrieval o S3 Glacier Deep Archive, paga el objeto archivado y la copia que restauró de manera temporal. Para obtener información acerca de los precios, consulte Precios de Amazon S3.

Restauración de objetos desde S3 Intelligent-Tiering

Cuando restaura un objeto desde el nivel Acceso a archivos de S3 Intelligent-Tiering o Acceso a archivos profundo de S3 Intelligent-Tiering, el objeto vuelve a entrar en el nivel Acceso frecuente de S3 Intelligent-Tiering. Si no se accede al objeto después de 30 días consecutivos, se mueve automáticamente al nivel de acceso poco frecuente. Después de un mínimo de 90 días consecutivos sin acceso, el objeto pasa a la capa de acceso de archivo de S3 Intelligent-Tiering. Si no se accede al objeto después de 180 días consecutivos, el objeto se mueve al nivel de acceso de archivo profundo.

> **Note**
>
> A diferencia de las clases de almacenamiento S3 Glacier Flexible Retrieval y S3 Glacier Deep Archive, las solicitudes de restauración para objetos de S3 Intelligent-Tiering no aceptan el valor `Days`.

Uso de operaciones por lotes de S3 con solicitudes de restauración

Para restaurar más de un objeto de Amazon S3 con una sola solicitud, puede utilizar Operaciones por lotes de S3. Proporcione a Operaciones por lotes de S3 una lista de objetos en los que operar. Operaciones por lotes de S3 llama a la operación de la API respectiva para realizar la operación especificada. Un solo trabajo de operaciones por lotes puede realizar la operación especificada en miles de millones de objetos con exabytes de datos.

Tiempo de restauración

Amazon S3 calcula la hora de vencimiento de la copia del objeto restaurada agregando el número de días especificados en la solicitud de restauración a la hora en que la restauración solicitada termina. Amazon S3 después redondea el tiempo resultante a la medianoche del día siguiente en hora universal coordinada (UTC). Por ejemplo, supongamos que el 15 de octubre de 2012 a las 10:30 UTC se crea una copia restaurada de un objeto y el periodo de restauración se fija en 3 días. En este caso, la copia restaurada tiene como fecha de vencimiento el 19 de octubre de 2012 a las 00:00 UTC, momento en el que Amazon S3 elimina la copia del objeto.

El tiempo que tarda un trabajo de restauración en finalizar depende de la clase de almacenamiento de archivo o el nivel de almacenamiento que utilice y la opción de recuperación que especifique: Acelerada (solo disponible para S3 Glacier Flexible Retrieval y Acceso a archivos de S3 Intelligent-

Tiering), Estándar o En bloque. Para obtener más información, consulte Opciones de recuperación de archivos.

Puede recibir una notificación cuando se haya completado la restauración mediante las notificaciones de eventos de Amazon S3. Para obtener más información, consulte Notificaciones de eventos de Amazon S3.

Temas

- Opciones de recuperación de archivos
- Restauración de un objeto archivado

Opciones de recuperación de archivos

A continuación, se muestran las opciones de recuperación disponibles al restaurar un objeto archivado en Amazon S3:

- Acelerada: acceda de manera rápida a los datos que se almacenan en la clase de almacenamiento S3 Glacier Flexible Retrieval o en el nivel Acceso a archivos de S3 Intelligent-Tiering. Puede utilizar esta opción cuando se requieran solicitudes urgentes ocasionales para un subconjunto de archivos. En todos los casos, excepto para los objetos archivados de mayor tamaño (más de 250 MB), los datos a los que se obtiene acceso mediante solicitudes rápidas suelen estar disponibles en un plazo de entre 1 y 5 minutos.

  > **Note**
  >
  > Las recuperaciones aceleradas son una característica premium que se cobra a la tarifa de la solicitud acelerada y de recuperación.
  > Para obtener información acerca de los precios de Amazon S3, consulte Precios de Amazon S3.

  La capacidad aprovisionada ayuda a garantizar que la capacidad de recuperación para las recuperaciones rápidas de S3 Glacier Flexible Retrieval esté disponible cuando la necesite. Para obtener más información, consulte Capacidad aprovisionada.
- Estándar: acceda a los objetos archivados en un plazo de varias horas. Estándar es la opción predeterminada para las solicitudes de recuperación que no especifican la opción de recuperación. Las recuperaciones estándar generalmente finalizan en un plazo de 3 a 5 horas para los objetos

almacenados en la clase de almacenamiento S3 Glacier Flexible Retrieval o en el nivel S3 Intelligent-Tiering Archive Access. Estas recuperaciones suelen finalizar en un plazo de 12 horas para los objetos almacenados en la clase de almacenamiento S3 Glacier Deep Archive o en el nivel S3 Intelligent-Tiering Deep Archive Access. Las recuperaciones estándar son gratuitas para los objetos que se almacenan en S3 Intelligent-Tiering.

> **Note**
>
> - Para objetos almacenados en la clase de almacenamiento S3 Glacier Flexible Retrieval o en nivel de Acceso a archivos de S3 Intelligent-Tiering, las recuperaciones estándar iniciadas mediante el uso de la operación de restauración de Operaciones por lotes de S3 suelen comenzar en cuestión de minutos y finalizar en un plazo de 3 a 5 horas.
> - Para objetos de la clase de almacenamiento S3 Glacier Deep Archive o del nivel Acceso a archivos profundos S3 Intelligent-Tiering, las recuperaciones estándar iniciadas mediante Operaciones por lotes suelen comenzar en un plazo de 9 horas y finalizar en 12 horas.

- En bloque: acceda a los datos mediante la opción de recuperación menos costosa de Amazon S3 Glacier. Con las recuperaciones en bloque, puede recuperar grandes cantidades de datos, incluso petabytes, de forma económica.

  Para los objetos que se almacenan en la clase de almacenamiento S3 Glacier Flexible Retrieval o en el nivel S3 Intelligent-Tiering Archive Access, las recuperaciones masivas suelen finalizar en un plazo de 5 a 12 horas. Para los objetos almacenados en la clase de almacenamiento S3 Glacier Deep Archive o en el nivel S3 Intelligent-Tiering Deep Archive Access, estas recuperaciones suelen finalizar en un plazo de 48 horas.

  Las recuperaciones masivas son gratuitas para los objetos que se almacenan en las clases de almacenamiento S3 Glacier Flexible Retrieval o S3 Intelligent-Tiering.

En la tabla siguiente se resumen las opciones de recuperación de archivos. Para obtener información acerca de los precios, consulte Precios de Amazon S3.

Para realizar una recuperación `Expedited`, `Standard` o `Bulk`, establezca el elemento de solicitud `Tier` en la solicitud de operación de la API de REST RestoreObject en la opción que desee, o el equivalente en la AWS Command Line Interface (AWS CLI) o los SDK de AWS. Si ha adquirido capacidad aprovisionada, todas las recuperaciones rápidas se realizarán automáticamente con su capacidad aprovisionada.

Capacidad aprovisionada

La capacidad aprovisionada garantiza que la capacidad de recuperación para las recuperaciones rápidas de S3 Glacier Flexible Retrieval esté disponible cuando la necesite. Cada unidad de capacidad permite que se puedan realizar al menos tres recuperaciones rápidas cada 5 minutos y proporciona hasta 150 megabytes por segundo (MBps) de rendimiento de recuperación.

Si su carga de trabajo requiere un acceso de confianza y predecible a un subconjunto de sus datos en cuestión de minutos, debe considerar la posibilidad de adquirir capacidad de recuperación aprovisionada. Sin la capacidad aprovisionada, puede que no se acepten las recuperaciones rápidas durante los períodos de alta demanda. Si necesita obtener acceso a recuperaciones rápidas incondicionalmente, le recomendamos que adquiera capacidad de recuperación aprovisionada.

Las unidades de capacidad aprovisionada se asignan a una Cuenta de AWS. Por lo tanto, el solicitante de la recuperación acelerada de datos debe comprar la unidad de capacidad aprovisionada, no el propietario del bucket.

Puede adquirir capacidad aprovisionada con la consola de Amazon S3, la consola de Amazon S3 Glacier, la operación de la API de REST Comprar capacidad aprovisionada, los SDK de AWS o la AWS CLI. Para obtener información acerca de los precios de la capacidad aprovisionada, consulte Precios de Amazon S3.

Tasas de solicitud de inicio de restauración de S3 Glacier

Cuando inicia solicitudes de restauración que están almacenados en las clases de almacenamiento S3 Glacier Flexible Retrieval o S3 Glacier Deep Archive, se le aplica una cuota de solicitudes de recuperación para la Cuenta de AWS. S3 Glacier admite solicitudes de restauración a una velocidad de hasta 1000 transacciones por segundo. Si se supera esta tasa, las solicitudes válidas se limitan o rechazan y Amazon S3 devuelve un error `ThrottlingException`.

De forma opcional, también puede utilizar las operaciones por lotes de S3 para recuperar una gran cantidad de objetos almacenados en S3 Glacier Flexible Retrieval o S3 Glacier Deep Archive con una sola solicitud. Para obtener más información, consulte Realización de operaciones por lotes a gran escala en objetos de Amazon S3.

Restauración de un objeto archivado

Los objetos de Amazon S3 de las siguientes clases o niveles de almacenamiento se archivan y no son accesibles en tiempo real:

- La clase de almacenamiento S3 Glacier Flexible Retrieval
- La clase de almacenamiento S3 Glacier Deep Archive
- La capa de acceso de archivo de S3 Intelligent-Tiering
- El nivel S3 Intelligent-Tiering Deep Archive Access

Los objetos de Amazon S3 almacenados en las clases de almacenamiento S3 Glacier Flexible Retrieval o S3 Glacier Deep Archive no son accesibles inmediatamente. Para tener acceso a un objeto en estas clases de almacenamiento, se debe restaurar una copia temporal del objeto en el bucket de S3 correspondiente durante un periodo especificado (número de días). Si desea obtener una copia permanente del objeto, restaure el objeto y, a continuación, cree una copia del objeto en su bucket de Amazon S3. La consola de Amazon S3 no admite la copia de objetos restaurados. Para este tipo de operación de copia, utilice la AWS Command Line Interface (AWS CLI), los SDK de AWS o la API de REST. A menos que haga una copia y cambie su clase de almacenamiento, el objeto seguirá almacenándose en las clases de almacenamiento S3 Glacier Flexible Retrieval o S3 Glacier Deep Archive. Para obtener información sobre el uso de estas clases de almacenamiento, consulte Clases de almacenamiento para objetos a los que se accede con poca frecuencia.

Para acceder a objetos de los niveles Acceso a archivos o Acceso a archivos profundo de S3 Intelligent-Tiering, se debe iniciar una solicitud de restauración y esperar hasta que el objeto se mueva al nivel Acceso frecuente. Cuando se restaura desde las capas de acceso de archivo de o acceso profundo, el objeto vuelve a pasar a la capa de acceso frecuente. Para obtener información sobre el uso de estas clases de almacenamiento, consulte Clase de almacenamiento para optimizar automáticamente los datos con patrones de acceso cambiantes o desconocidos.

Para obtener información general acerca de los objetos archivados, consulte Trabajar con objetos archivados.

> **Note**
>
> - Cuando restaura un archivo archivado desde las clases de almacenamiento S3 Glacier Flexible Retrieval o S3 Glacier Deep Archive, paga el objeto archivado y la copia que restauró de manera temporal.
> - Al restaurar un objeto desde S3 Intelligent-Tiering, las recuperaciones estándar o en bloque no conllevan gastos de recuperación.

- Las solicitudes de restauración posteriores llamadas en objetos archivados que ya se han restaurado se facturarán como una solicitud GET. Para obtener información acerca de los precios, consulte Precios de Amazon S3.

Restauración de un objeto archivado

Puede restaurar un objeto archivado mediante la consola de Amazon S3, la API de REST de Amazon S3, los SDK de AWS, la AWS Command Line Interface (AWS CLI) u Operaciones por lotes de S3.

Uso de la consola de S3

Restaurar objetos mediante la consola de Amazon S3

Utilice el siguiente procedimiento para Restaurar un objeto que se ha archivado en las clases de almacenamiento S3 Glacier Flexible Retrieval o S3 Glacier Deep Archive, o en los niveles de almacenamiento Acceso a archivos o Acceso a archivos profundo de S3 Intelligent-Tiering.

Para restaurar un objeto archivado

1. Inicie sesión en la AWS Management Console y abra la consola de Amazon S3 en https://console.aws.amazon.com/s3/.
2. En el panel de navegación izquierdo, elija Instancias.
3. En la lista Buckets (Buckets), seleccione el nombre del bucket que contiene los objetos que desea restaurar.
4. En la lista Name (Nombre), seleccione el objeto o los objetos que desea restaurar, elija Actions (Acciones) y, luego, Initiate restore (Iniciar restauración).
5. Si va a restaurar desde S3 Glacier Flexible Retrieval o S3 Glacier Deep Archive, introduzca el número de días que desea que sus datos archivados estén accesibles en el cuadro Número de días que la copia restaurada está disponible.
6. En Nivel de recuperación, realice una de las acciones siguientes:

   - Seleccione Recuperación en bloque o Recuperación estándar y, a continuación, seleccione Iniciar restauración.
   - Elija Expedited retrieval (Recuperación urgente) (disponible solo para S3 Glacier Flexible Retrieval o S3 Intelligent-Tiering Archive Access). Si va a restaurar un objeto en S3 Glacier Flexible Retrieval, puede elegir si desea adquirir capacidad aprovisionada para su

recuperación rápida. Si desea adquirir capacidad aprovisionada, continúe con el siguiente paso. En caso contrario, elija Iniciar restauración.

> **Note**
>
> Los objetos de los niveles Acceso a archivos y Acceso a archivos profundo de S3 Intelligent-Tiering se restauran automáticamente en el nivel Acceso frecuente.

7. (Opcional) Si va a restaurar un objeto en S3 Glacier Flexible Retrieval y ha elegido Recuperación rápida, puede elegir si desea adquirir capacidad aprovisionada. La capacidad aprovisionada solo está disponible para objetos en S3 Glacier Flexible Retrieval. Si ya tiene capacidad aprovisionada, seleccione Iniciar restauración para comenzar una recuperación aprovisionada.

   Si tiene capacidad aprovisionada, todas sus recuperaciones rápidas funcionan con la capacidad aprovisionada. Para obtener más información, consulte Capacidad aprovisionada.

   - Si no tiene capacidad aprovisionada y no desea comprarla, seleccione Iniciar restauración.
   - Si no tiene capacidad aprovisionada, pero quiere comprar unidades de capacidad aprovisionada (PCU), elija Adquirir PCU. En el cuadro de diálogo Adquirir PCU, elija cuántas PCU desea comprar, confirme la compra y, a continuación, elija Adquirir PCU. Cuando recibe el mensaje Compra realizada correctamente, seleccione Iniciar restauración para comenzar la recuperación aprovisionada.

Uso de la AWS CLI

Restaurar objetos desde S3 Glacier Flexible Retrieval o S3 Glacier Deep Archive

El siguiente ejemplo utiliza el comando `restore-object` para restaurar el objeto *`dir1/example.obj`* en el bucket *`amzn-s3-demo-bucket`* durante 25 días.

```
aws s3api restore-object --bucket amzn-s3-demo-bucket --key dir1/example.obj --restore-request '{"Days":25,"GlacierJobParameters":{"Tier":"Standard"}}'
```

Si la sintaxis JSON utilizada en el ejemplo produce un error en un cliente de Windows, reemplace la solicitud de restauración por la siguiente sintaxis:

```
--restore-request Days=25,GlacierJobParameters={"Tier"="Standard"}
```

Restaurar objetos desde Acceso a archivos y Acceso a archivos profundo de S3 Intelligent-Tiering

El siguiente ejemplo utiliza el comando `restore-object` para restaurar el objeto *dir1/example.obj* del bucket *amzn-s3-demo-bucket* en el nivel Acceso frecuente.

```
aws s3api restore-object --bucket amzn-s3-demo-bucket --key dir1/example.obj --restore-request '{}'
```

> **Note**
>
> A diferencia de las clases de almacenamiento S3 Glacier Flexible Retrieval y S3 Glacier Deep Archive, las solicitudes de restauración para objetos de S3 Intelligent-Tiering no aceptan el valor `Days`.

Monitorear el estado de restauración

Para monitorear el estado de la solicitud `restore-object`, use el comando `head-object` siguiente:

```
aws s3api head-object --bucket amzn-s3-demo-bucket --key dir1/example.obj
```

Para obtener más información, consulte restore-object en la Referencia de los comandos de AWS CLI.

Uso de la API de REST

Amazon S3 le proporciona una operación de la API para que pueda iniciar la restauración de un objeto archivado. Para obtener más información, consulte RestoreObject en la Referencia de la API de Amazon Simple Storage Service.

Uso de los AWS SDK

Para conocer ejemplos de cómo restaurar objetos archivados en S3 Glacier Flexible Retrieval o S3 Glacier Deep Archive con los SDK de AWS, consulte Uso de `RestoreObject` con un AWS SDK o la CLI.

Uso de Operaciones por lotes de S3

Para restaurar más de un objeto archivado con una sola solicitud, puede utilizar Operaciones por lotes de S3. Proporcione a Operaciones por lotes de S3 una lista de objetos en los que operar.

Operaciones por lotes de S3 llama a la operación de la API respectiva para realizar la operación especificada. Un solo trabajo de operaciones por lotes puede realizar la operación especificada en miles de millones de objetos con exabytes de datos.

Para crear un trabajo de Operaciones por lotes, debe disponer de un manifiesto que contenga solo los objetos que desea restaurar. Puede crear un manifiesto mediante Inventario de S3 o puede proporcionar un archivo CSV con la información necesaria. Para obtener más información, consulte the section called “Especificar un manifiesto”.

Antes de crear y ejecutar trabajos de Operaciones por lotes de S3, debe conceder permisos a Amazon S3 para que realice Operaciones por lotes de S3 en su nombre. Para conocer los permisos necesarios, consulte the section called “Concesión de permisos”.

> **Note**
>
> Los trabajos de Operaciones por lotes pueden operar en objetos de clase de almacenamiento de S3 Glacier Flexible Retrieval y S3 Glacier Deep Archive o en objetos de nivel de almacenamiento Acceso a archivos o Acceso a archivos profundo de S3 Intelligent-Tiering. Operaciones por lotes no pueden funcionar con ambos tipos de objetos archivados en el mismo trabajo. Para restaurar objetos de ambos tipos, debe crear trabajos de operaciones por lotes independientes.
> Para obtener más información sobre cómo utilizar Operaciones por lotes para restaurar objetos archivados, consulte the section called “Restaurar objetos”.

Para crear un trabajo de Operaciones por lotes de S3 para iniciar la restauración de objetos

1. Inicie sesión AWS Management Console Management Console y abra la consola de Amazon S3 en https://console.aws.amazon.com/s3/.
2. En el panel de navegación izquierdo, elija Batch Operations (Operaciones por lote).
3. Seleccione Crear trabajo.
4. Para Región de AWS, elija la región en la que desea crear el trabajo.
5. En Formato del manifiesto, seleccione el tipo de manifiesto que desee usar.

   - Si elige Informe de inventario de S3, introduzca la ruta al objeto `manifest.json` que Amazon S3 generó como parte del informe de inventario con formato CSV. Si desea utilizar una versión del manifiesto distinta de la más reciente, introduzca el ID de versión del objeto `manifest.json`.

- Si selecciona CSV, escriba la ruta del objeto del manifiesto con formato CSV. El objeto del manifiesto debe tener el mismo formato que se ha especificado en la consola. Si desea utilizar una versión distinta de la más reciente, puede incluir opcionalmente el ID de versión del objeto de manifiesto.

6. Elija Siguiente.
7. En la sección Operación, elija Restaurar.
8. En la sección Restaurar, para Origen de restauración, elija Glacier Flexible Retrieval o Glacier Deep Archive o Nivel Acceso a archivos o Acceso a archivos profundo de Intelligent-Tiering.

   Si eligió Glacier Flexible Retrieval o Glacier Deep Archive, introduzca un número en Número de días que está disponible la copia restaurada.

   En Nivel de recuperación, elija el nivel que desee utilizar.
9. Elija Siguiente.
10. En la página Configurar opciones adicionales, rellene las siguientes secciones:

    - En la sección Opciones adicionales, proporcione una descripción para el trabajo y especifique un número de prioridad para el trabajo. Los números más altos indican una mayor prioridad. Para obtener más información, consulte the section called “Asignar prioridad a los trabajos”.
    - En la sección Informe de finalización, seleccione si Operaciones por lotes debe crear un informe de finalización. Para obtener más información sobre los informes de finalización, consulte the section called “Informes de finalización”.
    - En la sección Permisos, debe conceder permisos a Amazon S3 para que ejecute Operaciones por lotes en su nombre. Para conocer los permisos necesarios, consulte the section called “Concesión de permisos”.
    - (Opcional) En la sección Etiquetas de trabajo, agregue etiquetas en pares clave-valor. Para obtener más información, consulte the section called “Uso de etiquetas”.

    Cuando haya terminado, elija Siguiente.
11. En la página Review (Revisar), puede verificar las configuraciones. Si necesita realizar cambios, seleccione Anterior. De lo contrario, seleccione Crear trabajo.

Para obtener más información sobre Operaciones por lotes, consulte Restaurara objetos con Operaciones por lotes y Creación de trabajos de operaciones por lotes de S3.

Comprobación del estado de restauración y de la fecha de vencimiento

Puede consultar el estado de una solicitud de restauración o la fecha de vencimiento mediante la consola de Amazon S3, las notificaciones de eventos de Amazon S3, la AWS CLI o la API de REST de Amazon S3.

> **Note**
>
> Los objetos restaurados de las clases de almacenamiento S3 Glacier Flexible Retrieval y S3 Glacier Deep Archive se almacenan solo durante el número de días que especifique. Los siguientes procedimientos devuelven la fecha de vencimiento de estas copias.
> Los objetos restaurados desde los niveles de almacenamiento S3 Intelligent-Tiering Archive Access y Deep Archive Access no tienen fecha de vencimiento y se trasladan de nuevo al nivel Acceso frecuente.

Uso de la consola de S3

Para comprobar el estado de restauración y la fecha de caducidad de un objeto en la consola de Amazon S3

1. Abra la consola de Amazon S3 en https://console.aws.amazon.com/s3/.
2. En el panel de navegación situado a la izquierda, elija Buckets.
3. En la lista Buckets, seleccione el nombre del bucket que contiene el objeto que va a restaurar.
4. En la lista Objetos, seleccione el objeto que va a restaurar. Aparece la página de detalles del objeto.

   - Si la restauración no ha finalizado, en la parte superior de la página verá una sección que indica Restauración en curso.
   - Si la restauración ha finalizado, en la parte superior de la página verá una sección que indica Restauración completa. Si restaura desde S3 Glacier Flexible Retrieval o S3 Glacier Deep Archive, esta sección también muestra la Fecha de caducidad de la restauración. Amazon S3 eliminará la copia restaurada de su objeto archivado en esta fecha.

Uso de notificaciones de eventos de Amazon S3

Puede recibir notificaciones sobre la finalización de la restauración del objeto mediante el uso de la acción `s3:ObjectRestore:Completed` con la característica Notificaciones de eventos de Amazon

S3. Para obtener más información sobre cómo habilitar las notificaciones de eventos, consulte Habilitación de notificaciones de Amazon SQS, Amazon SNS y AWS Lambda. Para obtener más información acerca de los diferentes tipos de eventos `ObjectRestore`, consulte the section called "Tipos de eventos admitidos para SQS, SNS y Lambda".

Uso de la AWS CLI

Comprobar el estado de restauración y la fecha de caducidad de un objeto con la AWS CLI

En el siguiente ejemplo, se utiliza el comando `head-object` para ver los metadatos del objeto *`dir1/example.obj`* en el bucket *`amzn-s3-demo-bucket`*. Cuando ejecuta este comando en un objeto que se está restaurando, Amazon S3 devuelve si la restauración está en curso y (si procede) la fecha de caducidad.

```
aws s3api head-object --bucket amzn-s3-demo-bucket --key dir1/example.obj
```

Resultado previsto (restauración en curso):

```
{
    "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"
}
```

Resultado previsto (restauración finalizada):

```
{
    "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"
}
```

Para obtener más información sobre `head-object`, consulte head-object en la Referencia de los comandos de AWS CLI.

Uso de la API de REST

Amazon S3 proporciona una operación de la API para recuperar los metadatos de objeto. Para comprobar el estado de restauración y la fecha de caducidad de un objeto archivado mediante la API de REST, consulte HeadObject en la Referencia de la API de Amazon Simple Storage Service.

Actualizar la velocidad de una restauración en curso

Puede actualizar la velocidad de la restauración mientras esta se encuentra en curso.

Para actualizar una restauración en curso a una capa más rápida

1. Abra la consola de Amazon S3 en https://console.aws.amazon.com/s3.
2. En el panel de navegación situado a la izquierda, elija Buckets.
3. En la lista Buckets (Buckets), seleccione el nombre del bucket que contiene los objetos que desea restaurar.
4. En la lista Objetos, seleccione el objeto que va a restaurar. Aparece la página de detalles del objeto. En la página de detalles del objeto, elija Actualizar nivel de recuperación. Para obtener más información sobre cómo comprobar el estado de restauración de un objeto, consulte Comprobación del estado de restauración y de la fecha de vencimiento.
5. Seleccione el nivel al que desea realizar la actualización y elija Iniciar restauración.

Usar Bloqueo de objetos de S3

Bloqueo de objetos de S3 puede ayudar a evitar que se eliminen o se sobreescriban objetos de Amazon S3 durante un periodo de tiempo determinado o de manera indefinida. Bloqueo de objetos utiliza un modelo de escritura única y lectura múltiple (WORM) para almacenar objetos. Bloqueo de objetos se puede utilizar para cumplir con los requisitos normativos que requieren almacenamiento WORM o simplemente para agregar otra capa de protección para evitar cambios o eliminaciones de objetos.

> **Note**
>
> Cohasset Associates ha evaluado el uso de Bloqueo de objetos de S3 en entornos sujetos a las normativas SEC 17a-4, CFTC y FINRA. Para obtener más información acerca de cómo Bloqueo de objetos está relacionado con estas regulaciones, consulte la Evaluación de cumplimiento de Cohasset Associates.

Bloqueo de objetos de S3 ofrece dos maneras de administrar la retención de objetos: periodos de retención y retenciones legales. Una versión de un objeto puede tener un periodo de retención, una retención legal, o ambos.

- Periodo de retención: un periodo de retención especifica un periodo de tiempo fijo durante el cual los objetos permanecen bloqueados. También puede establecer un periodo de retención único para los objetos individuales. Además, puede establecer un periodo de retención predeterminado en un bucket de S3. También puede restringir los períodos de retención mínimos y máximos permitidos con la clave de condición `s3:object-lock-remaining-retention-days` de la política de bucket. Esto lo ayuda a establecer un rango de períodos de retención y a restringir los períodos de retención que pueden ser más cortos o más largos que este rango.
- Retención legal: una retención legal proporciona la misma protección que un periodo de retención, pero no tiene fecha de vencimiento. En cambio, la retención legal sigue vigente hasta que la elimine explícitamente. Las retenciones legales son independientes de los periodos de retención y se aplican a versiones de objetos individuales.

El bloqueo de objetos solo funciona en buckets que tienen activado el control de versiones de S3. Cuando bloquea una versión de un objeto, Amazon S3 almacena la información de bloqueo en los metadatos de esa versión del objeto. Al colocar un periodo de retención o retención legal en un objeto se protege solo la versión que se especifica en la solicitud. Los periodos de retención y las retenciones legales no impiden la creación de nuevas versiones del objeto ni la adición de marcadores de borrado encima del objeto. Para obtener más información sobre el control de versiones de S3, consulte Usar el control de versiones en buckets de S3.

Si pone un objeto en un bucket que ya contiene un objeto que ya existe y está protegido con el mismo nombre de clave del objeto, Amazon S3 crea una nueva versión de ese objeto. La versión existente y protegida del objeto permanece bloqueada en función de su configuración de retención.

Cómo funciona Bloqueo de objetos de S3

Temas

Periodos de retención

Un periodo de retención protege una versión de un objeto por un periodo de tiempo determinado. Cuando coloca un periodo de retención en una versión del objeto, Amazon S3 almacena una marca temporal en los metadatos de la versión del objeto para indicar el vencimiento del periodo de retención. Cuando vence el periodo de retención, es posible sobrescribir o eliminar la versión del objeto.

Puede colocar un periodo de retención de forma explícita en la versión de un objeto individual o en las propiedades de un bucket para que se aplique automáticamente a todos los objetos del bucket. Cuando se aplica un periodo de retención a una versión de objeto explícitamente, se especifica una Fecha de finalización de la retención de la versión de objeto. Amazon S3 almacena esta fecha en los metadatos de la versión del objeto.

También puede establecer un periodo de retención en las propiedades de un bucket. Al establecer un periodo de retención en un bucket, especifica una duración, en días o años, para proteger cada versión del objeto colocada en el bucket. Cuando coloca un objeto en un bucket, Amazon S3 calcula una fecha de finalización de la retención de la versión del objeto agregando la duración especificada a la marca temporal de creación de la versión del objeto. La versión del objeto queda protegida exactamente como si hubiese colocado en esta un bloqueo individual con ese periodo de retención.

> **Note**
>
> Cuando se aplica PUT a la versión de un objeto que tiene un modo y un periodo de retención individual explícitos en un bucket, la configuración de bloqueo de objetos individual de la versión del objeto anula cualquier configuración de retención para las propiedades del bucket.

Como todos los demás ajustes de Bloqueo de objetos, los periodos de retención solo se aplican a versiones del objeto individuales. Diferentes versiones del mismo objeto pueden tener distintos modos y periodos de retención.

Por ejemplo, supongamos que tiene un objeto para el que han transcurrido 15 días de un periodo de retención de 30 días y aplica PUT a un objeto en Amazon S3 con el mismo nombre y un periodo de retención de 60 días. En este caso, su solicitud PUT se considerará correcta y Amazon S3 creará una nueva versión del objeto con un periodo de retención de 60 días. La versión antigua mantiene su periodo de retención original y podrá borrarse en 15 días.

Una vez que ha aplicado una configuración de retención a una versión de un objeto, puede ampliar el periodo de retención. Para ello, envíe una nueva solicitud de bloqueo de objetos para la versión del objeto con una Fecha de finalización de la retención que es posterior a la configurada actualmente para esa versión del objeto. Amazon S3 reemplaza el periodo de retención actual con el periodo nuevo, más largo. Cualquier usuario con permisos para colocar un periodo de retención en un objeto puede ampliar un periodo de retención para una versión del objeto. Para establecer un periodo de retención, debe tener el permiso `s3:PutObjectRetention`.

Al establecer un periodo de retención en un objeto o en un bucket de S3, debe seleccionar uno de los dos modos de retención: cumplimiento o gobierno.

Modos de retención

Bloqueo de objetos de S3 brinda dos modos de retención que aplican diferentes niveles de protección a los objetos:

- Modo Cumplimiento
- Modo Gobierno

En el modo Conformidad, ningún usuario puede sobrescribir ni eliminar una versión de objeto protegida, incluido el usuario raíz de la Cuenta de AWS. Una vez que se ha bloqueado un objeto en el modo Cumplimiento, no es posible cambiar su modo de retención ni acortar su periodo de retención. El modo de conformidad evita que se pueda sobrescribir o eliminar una versión del objeto durante toda la duración del periodo de retención.

> **Note**
>
> La única forma de eliminar un objeto en el modo de cumplimiento antes de que venza su fecha de retención es eliminar la Cuenta de AWS asociada.

En el modo Gobierno, los usuarios no pueden sobrescribir ni eliminar una versión del objeto ni alterar su configuración de bloqueo a menos que tengan permisos especiales. Con el modo Gobierno, evita que la mayoría de los usuarios eliminen un objeto, pero puede seguir otorgando permiso a algunos usuarios para alterar la configuración de retención o eliminar los objetos si es necesario. También puede usar el modo Gobierno para probar la configuración del periodo de retención antes de crear un periodo de retención en el modo Cumplimiento.

Para anular o eliminar la configuración de retención del modo Gobierno, debe tener el permiso `s3:BypassGovernanceRetention` que debe incluir explícitamente `x-amz-bypass-governance-retention:true` como encabezado de la solicitud con cualquier solicitud que requiera la anulación del modo Gobierno.

> **Note**
>
> De forma predeterminada, la consola de Amazon S3 incluye el encabezado `x-amz-bypass-governance-retention:true`. Si intenta eliminar objetos protegidos por el modo Gobierno y cuenta con los permisos `s3:BypassGovernanceRetention`, la operación se realizará correctamente.

Retenciones legales

Con Bloqueo de objetos de S3 también puede colocar una retención legal en una versión de objeto. Al igual que un periodo de retención, la retención legal impide que se sobrescriba o elimine una versión de un objeto. Sin embargo, una retención legal no tiene una cantidad de tiempo asociada y sigue vigente hasta que se elimine. Los usuarios con el permiso `s3:PutObjectLegalHold` pueden colocar y eliminar libremente retenciones legales.

Las retenciones legales son independientes de los periodos de retención. La colocación de una retención legal en una versión de un objeto no afecta al modo ni al periodo de retención de dicha versión del objeto.

Por ejemplo, supongamos que coloca una retención legal en una versión de un objeto y que esta también recibe protección de un periodo de retención. Si vence el periodo de retención, el objeto no perderá su protección WORM. En cambio, la retención legal continúa protegiendo el objeto hasta que un usuario autorizado la elimine explícitamente. De igual modo, si elimina una retención legal en una versión del objeto que tiene un periodo de retención vigente, la versión del objeto seguirá protegida hasta que venza el periodo de retención.

Prácticas recomendadas para el uso de Bloqueo de objetos de S3

Considere la posibilidad de utilizar el Modo Gobernanza si desea evitar que la mayoría de los usuarios eliminen los objetos durante un período de retención predefinido, pero al mismo tiempo quiere que algunos usuarios con permisos especiales tengan la flexibilidad de modificar la configuración de retención o eliminar los objetos.

Considere la posibilidad de utilizar el Modo Gobernanza si no quiere que ningún usuario, ni siquiera el usuario raíz de su Cuenta de AWS, pueda eliminar los objetos durante un período de retención predefinido. Puede utilizar este modo en caso de que necesite almacenar datos conformes.

Puede utilizar Retención legal si no sabe exactamente cuánto tiempo desea que sus objetos permanezcan inmutables. Esto puede deberse a que se va a realizar una auditoría externa de los datos próximamente y desea mantener los objetos inmutables hasta que finalice la auditoría. O podría tener un proyecto en curso que utiliza un conjunto de datos que desea mantener inmutables hasta que se complete el proyecto.

Permisos necesarios

Las operaciones de Bloqueo de objetos necesitan permisos específicos. En función de la operación exacta que esté intentando, es posible que necesite alguno de los siguientes permisos:

- `s3:BypassGovernanceRetention`
- `s3:GetBucketObjectLockConfiguration`
- `s3:GetObjectLegalHold`
- `s3:GetObjectRetention`
- `s3:PutBucketObjectLockConfiguration`
- `s3:PutObjectLegalHold`
- `s3:PutObjectRetention`

Para obtener una lista completa de los permisos de S3 con descripciones, consulte Actions, resources, and condition keys for Amazon S3 en la Referencia de autorizaciones de servicio.

Para obtener información sobre el uso de condiciones con permisos, consulte Ejemplos de políticas de bucket que utilizan claves de condición.

Consideraciones sobre el bloqueo de objetos

Bloqueo de objetos de S3 puede ayudar a evitar que se eliminen o se sobrescriban objetos durante un periodo de tiempo determinado o de manera indefinida.

Puede utilizar la consola de Amazon S3, AWS Command Line Interface (AWS CLI), los SDK de AWS o la API de REST de Amazon S3 para ver o configurar la información de Bloqueo de objetos. Para obtener información general acerca de las capacidades de Bloqueo de objetos de S3, consulte Usar Bloqueo de objetos de S3.

> **Important**
>
> - Una vez que active el bloqueo de objetos en un bucket, no puede desactivarlo ni suspender el control de versiones en ese bucket.
> - Los buckets de S3 con bloqueo de objetos no se pueden utilizar como buckets de destino para los registros de acceso al servidor. Para obtener más información, consulte the section called “Registro de acceso al servidor”.

Temas

- Establecer límites a los periodos de retención con una política de bucket

Permisos para ver la información de bloqueo

Puede ver mediante programación el estado de bloqueo de objetos de una versión del objeto de Amazon S3 mediante las operaciones HeadObject o GetObject. Ambas operaciones devuelven el modo de retención, la fecha límite de retención y el estado de la retención legal de la versión del objeto especificada. Además, puede ver el estado de Bloqueo de objetos de varios objetos de su bucket de S3 mediante Inventario de S3.

A fin de ver el periodo de retención y el modo de retención de la versión del objeto, debe tener el permiso `s3:GetObjectRetention`. A fin de ver el estado de la retención legal de la versión del objeto, debe tener el permiso `s3:GetObjectLegalHold`. Para ver la configuración de retención predeterminada del bucket, debe tener el permiso `s3:GetBucketObjectLockConfiguration`. Si realiza una solicitud para ver la configuración de bloqueo de objetos en un bucket que no tiene activado Bloqueo de objetos de S3, Amazon S3 devuelve un error.

Omitir el modo de gobierno

Si tiene el permiso `s3:BypassGovernanceRetention`, puede realizar operaciones en versiones de objetos bloqueados en modo Gobierno como si no estuvieran protegidos. Estas operaciones incluyen eliminar una versión del objeto, acortar el periodo de retención o eliminar el periodo de retención de Bloqueo de objetos colocando una nueva solicitud `PutObjectRetention` con parámetros vacíos.

Para omitir el modo Gobierno, debe indicar explícitamente en la solicitud que desea omitir este modo. Para ello, incluya el encabezado `x-amz-bypass-governance-retention:true` en la solicitud de operación de la API `PutObjectRetention` o utilice el parámetro equivalente en las solicitudes realizadas por medio de la AWS CLI o los SDK de AWS. La consola de S3 aplica automáticamente este encabezado a las solicitudes realizadas a través de la consola de S3 si tiene el permiso `s3:BypassGovernanceRetention`.

> **Note**
>
> Omitir el modo Gobierno no afecta al estado de retención legal de la versión del objeto. Si una versión de un objeto tiene habilitada una retención legal, esta se mantiene e impide solicitudes de sobrescribir o eliminar la versión del objeto.

Uso de Bloqueo de objetos con la replicación de S3

Puede utilizar el bloqueo de objetos con la replicación de S3 para permitir copias asíncronas y automáticas de objetos bloqueados y sus metadatos de retención en los buckets de S3. Esto significa que, en el caso de los objetos replicados, Amazon S3 adopta la configuración de bloqueo de objetos del bucket de origen. Dicho de otro modo, si el bucket de origen tiene habilitado el bloqueo de objetos, los buckets de destino deben tenerlo también habilitado. Si un objeto se carga directamente en el bucket de destino (fuera de la replicación de S3), utiliza el bloqueo de objetos establecido en el bucket de destino. Cuando utiliza la replicación, los objetos de un bucket de origen se replican en uno o varios buckets de destino.

Para configurar la replicación en un bucket con el Bloqueo de objetos habilitado, puede utilizar la consola de S3, AWS CLI, la API de REST de Amazon S3 o los SDK de AWS.

> **Note**
>
> Para utilizar Bloqueo de objetos con replicación, debe conceder dos nuevos permisos en el bucket de S3 de origen del rol de AWS Identity and Access Management (IAM) que utiliza para configurar la replicación. Los dos nuevos permisos adicionales son `s3:GetObjectRetention` y `s3:GetObjectLegalHold`. Si el rol tiene una instrucción de permisos `s3:Get*`, esa instrucción cumple el requisito. Para obtener más información, consulte Configuración de permisos para la replicación en directo.
>
> Para obtener información general acerca de la replicación de S3, consulte Información general de la replicación de objetos.
>
> Para ver ejemplos de cómo configurar la replicación de S3, consulte Ejemplos para configurar la replicación en directo.

Uso de Bloqueo de objetos con Inventario de Amazon S3

Es posible configurar el Inventario de Amazon S3 para crear listas de objetos en un bucket de S3 en un periodo definido. Puede configurar el Inventario de Amazon S3 para que incluya los siguientes metadatos de Bloqueo de objetos para sus objetos:

- La fecha límite de retención
- El modo de retención
- El estado de retención legal

Para obtener más información, consulte Inventario de Amazon S3.

Administración de las políticas de S3 Lifecycle con Bloqueo de objetos

Las configuraciones de administración del ciclo de vida del objeto continúan funcionando normalmente en objetos protegidos, incluida la colocación de marcadores de eliminación. Sin embargo, una política de caducidad de S3 Lifecycle no puede eliminar una versión bloqueada de un objeto. El bloqueo de objetos se mantiene independientemente de la clase de almacenamiento en la que resida el objeto y durante las transiciones de S3 Lifecycle entre clases de almacenamiento.

Para obtener más información acerca de la administración de ciclos de vida de los objetos, consulte Administración del ciclo de vida del almacenamiento.

Administración de los marcadores de eliminación con Bloqueo de objetos

Aunque no puede eliminar una versión de objeto protegida, puede crear un marcador de eliminación para ese objeto. Cuando se coloca un marcador de eliminación en un objeto, el objeto no se elimina ni tampoco ninguna versión de este. Sin embargo, hace que Amazon S3 se comporte de muchas de las maneras que lo haría si el objeto se hubiese eliminado. Para obtener más información, consulte Trabajar con marcadores de eliminación.

> **Note**
>
> Los marcadores de eliminación no tienen protección WORM, independientemente del periodo de retención o retención legal que se haya aplicado al objeto subyacente.

Uso de S3 Storage Lens con bloqueo de objetos

Para ver las métricas de los bytes de almacenamiento habilitados para Object Lock y el recuento de objetos, puede utilizar la Lente de almacenamiento de Amazon S3. Lente de almacenamiento de S3 es una función de análisis de almacenamiento en la nube que puede utilizar para obtener visibilidad en toda la organización sobre el uso y la actividad del almacenamiento de objetos.

Para obtener más información, consulte Uso de S3 Storage Lens para proteger sus datos.

Para obtener una lista completa de métricas, consulte Glosario de métricas de Amazon S3 Storage Lens.

Descarga de objetos a un bucket habilitado para Bloqueo de objetos

El encabezado `Content-MD5` es obligatorio para cualquier solicitud de carga de un objeto con un período de retención configurado mediante Bloqueo de objetos. Una forma de verificar la integridad del objeto después de cargarlo en el bucket es proporcionar un resumen MD5. Tras cargar el objeto, Amazon S3 calcula el resumen MD5 del objeto y lo compara con el valor que proporcionó. La solicitud se realiza correctamente solo si los dos resúmenes coinciden. La consola S3 añade automáticamente este encabezado, pero debe especificarlo cuando utilice la API PutObject.

Para obtener más información, consulte Uso de Content-MD5 al cargar objetos.

Configurar eventos y notificaciones

Puede utilizar las notificaciones de eventos de Amazon S3 para realizar un seguimiento del acceso a los datos y las configuraciones del bloqueo de objetos y los cambios que sufren mediante AWS CloudTrail. Para obtener información acerca de CloudTrail, consulte ¿Qué es AWS CloudTrail? en la Guía del usuario de AWS CloudTrail.

También puede utilizar Amazon CloudWatch para generar alertas según estos datos. Para obtener información acerca de CloudWatch, consulte ¿Qué es Amazon CloudWatch? en la Guía del usuario de Amazon CloudWatch.

Establecer límites a los periodos de retención con una política de bucket

Puede establecer los periodos de retención mínimos y máximos permitidos para un bucket con una política de bucket. El periodo máximo de retención es de 100 años.

En el siguiente ejemplo se muestra una política de bucket que utiliza la clave de condición `s3:object-lock-remaining-retention-days` para establecer un periodo de retención máximo de 10 días.

```
{
    "Version": "2012-10-17",
    "Id": "SetRetentionLimits",
    "Statement": [
        {
            "Sid": "SetRetentionPeriod",
            "Effect": "Deny",
            "Principal": "*",
            "Action": [
                "s3:PutObjectRetention"
```

```
            ],
            "Resource": "arn:aws:s3:::amzn-s3-demo-bucket1/*",
            "Condition": {
                "NumericGreaterThan": {
                    "s3:object-lock-remaining-retention-days": "10"
                }
            }
        }
    ]
}
```

> **Note**
>
> Si el bucket es el bucket de destino para una configuración de replicación, puede configurar períodos de retención mínimos y máximos admisibles para réplicas de objeto que se crean utilizando la replicación. Para ello, debe permitir la acción `s3:ReplicateObject` en su política de bucket. Para obtener más información sobre los permisos de replicación, consulte the section called “Configuración de permisos”.

Para obtener más información acerca de las políticas de buckets, consulte los siguientes temas:

- Actions, resources, and condition keys for Amazon S3 en la Referencia de autorizaciones de servicio
- Operaciones con objetos
- Ejemplos de políticas de bucket que utilizan claves de condición

Configurar el Bloqueo de objetos de S3

El bloqueo de objetos de Amazon S3 le permite almacenar objetos en Amazon S3 con un modelo de escritura única y lectura múltiple (WORM). Puede usar el bloqueo de objetos de S3 para evitar que se elimine o se sobrescriba un objeto durante un periodo de tiempo determinado o de manera indefinida. Para obtener información general acerca de las capacidades de Bloqueo de objetos, consulte Usar Bloqueo de objetos de S3.

Antes de bloquear cualquier objeto, debe habilitar el control de versiones y el bloqueo de objetos de S3 en un bucket. Posteriormente, puede establecer un periodo de retención, una retención legal o ambas opciones.

Para trabajar con Bloqueo de objetos, debe contar con determinados permisos. Para obtener una lista de los permisos relacionados con diversas operaciones de bloqueo de objetos, consulte the section called “Permisos necesarios”.

> **Important**
>
> - Una vez que active el bloqueo de objetos en un bucket, no puede desactivarlo ni suspender el control de versiones en ese bucket.
> - Los buckets de S3 con bloqueo de objetos no se pueden utilizar como buckets de destino para los registros de acceso al servidor. Para obtener más información, consulte the section called “Registro de acceso al servidor”.

Temas

Activación de Bloqueo de objetos al crear un nuevo bucket de S3

Puede habilitar el bloqueo de objetos al crear un bucket de S3 nuevo a través de la consola de Amazon S3, AWS Command Line Interface (AWS CLI), los SDK de AWS o la API de REST de Amazon S3.

Uso de la consola de S3

1. Inicie sesión en la AWS Management Console y abra la consola de Amazon S3 en https://console.aws.amazon.com/s3/.
2. En el panel de navegación izquierdo, elija Instancias.
3. Elija Crear bucket.

   Se abrirá la página Crear bucket.
4. En Nombre del bucket, escriba un nombre para el bucket.

> **Note**
>
> Una vez que haya creado un bucket, no podrá modificar su nombre. Para obtener más información sobre la nomenclatura de los buckets, consulte Reglas de nomenclatura de buckets.

5. En Región, elija la Región de AWS en la que desea que se encuentre el bucket.
6. En Propiedad de objetos, elija esta opción para desactivar o habilitar las listas de control de acceso (ACL) y controlar la propiedad de los objetos cargados en el bucket:
7. En Configuración de bloqueo de acceso público para este bucket, elija la configuración Bloquear acceso público que desee aplicar al bucket.
8. En Versiones del bucket, elija Editar.

   El Bloqueo de objetos solo funciona con buckets versionados.
9. (Opcional) En Tags (Etiquetas), puede elegir añadir etiquetas a su bucket. Las etiquetas son pares clave-valor que se utilizan para categorizar el almacenamiento y asignar los costes.
10. En Configuración avanzada, busque Bloqueo de objetos y elija Habilitar.

    Debe reconocer que si habilita el bloqueo de objetos, se bloquearán permanentemente los objetos de este bucket.
11. Elija Crear bucket.

Uso de la AWS CLI

En el siguiente ejemplo `create-bucket` se crea un nuevo bucket de S3 denominado *amzn-s3-demo-bucket1* con Bloqueo de objetos activado:

```
aws s3api create-bucket --bucket amzn-s3-demo-bucket1 --object-lock-enabled-for-bucket
```

Para obtener más información y ejemplos, consulte create-bucket en la referencia de comandos de AWS CLI.

> **Note**
>
> Puede ejecutar comandos de AWS CLI desde la consola mediante AWS CloudShell. AWS CloudShell es un intérprete de comandos previamente autenticado y basado en el navegador

que se puede lanzar directamente desde la página web de la AWS Management Console. Para obtener más información, consulte ¿Qué es CloudShell? en la Guía del usuario de AWS CloudShell.

Uso de la API de REST

Puede utilizar la API de REST para crear un bucket de S3 nuevo con Bloqueo de objetos activado. Para obtener más información, consulte CreateBucket en la Referencia de la API de Amazon Simple Storage Service.

Uso de los AWS SDK

Para ver ejemplos de cómo habilitar el bloqueo de objetos al crear un nuevo bucket de S3 con los SDK de AWS, consulte Uso de `CreateBucket` con un AWS SDK o la CLI.

Para ver ejemplos de cómo obtener la configuración actual de bloqueo de objetos con los SDK de AWS, consulte, Uso de `GetObjectLockConfiguration` con un AWS SDK o la CLI.

Para ver un escenario interactivo que muestre diferentes características de bloqueo de objetos mediante AWS SDK, consulte Trabajo con las características de bloqueo de objetos de Amazon S3 mediante un SDK de AWS.

Para obtener información general sobre el uso de diferentes SDK de AWS, consulte Desarrollo con Amazon S3 mediante los SDK de AWS.

Activación de Bloqueo de objetos en un bucket de S3 existente

Puede habilitar el bloqueo de objetos para un bucket de S3 existente a través de la consola de Amazon S3, la AWS CLI, los SDK de AWS o la API de REST de Amazon S3.

Uso de la consola de S3

> **Note**
>
> El Bloqueo de objetos solo funciona con buckets versionados.

1. Inicie sesión AWS Management Console Management Console y abra la consola de Amazon S3 en https://console.aws.amazon.com/s3/.

2. En el panel de navegación izquierdo, elija Instancias.
3. En la lista Buckets, elija el nombre del bucket para el que desea activar el Bloqueo de objetos de S3.
4. Elija la pestaña Propiedades.
5. En Propiedades, desplácese hacia abajo hasta la sección Bloqueo de objetos y seleccione Editar.
6. En Bloqueo de objetos, seleccione Habilitar.

   Debe reconocer que si habilita el bloqueo de objetos, se bloquearán permanentemente los objetos de este bucket.
7. Elija Guardar cambios.

Uso de la AWS CLI

El siguiente comando de ejemplo `put-object-lock-configuration` establece un período de retención de 50 días para el Bloqueo de objetos en un bucket denominado *amzn-s3-demo-bucket1*:

```
aws s3api put-object-lock-configuration --bucket amzn-s3-demo-bucket1 --object-
lock-configuration='{ "ObjectLockEnabled": "Enabled", "Rule": { "DefaultRetention":
 { "Mode": "COMPLIANCE", "Days": 50 }}}'
```

Para obtener más información y ejemplos, consulte put-object-lock-configuration en la referencia de comandos de AWS CLI.

> **Note**
>
> Puede ejecutar comandos de AWS CLI desde la consola mediante AWS CloudShell. AWS CloudShell es un intérprete de comandos previamente autenticado y basado en el navegador que se puede lanzar directamente desde la página web de la AWS Management Console. Para obtener más información, consulte ¿Qué es CloudShell? en la Guía del usuario de AWS CloudShell.

Uso de la API de REST

Puede utilizar la API de REST de Amazon S3 para habilitar el bloqueo de objetos en un bucket de S3 existente. Para obtener más información, consulte PutObjectLockConfiguration en la Referencia de la API de Amazon Simple Storage Service.

Uso de los AWS SDK

Para ver ejemplos de cómo habilitar el bloqueo de objetos para un bucket de S3 existente con los SDK de AWS, consulte Uso de `PutObjectLockConfiguration` con un AWS SDK o la CLI.

Para ver ejemplos de cómo obtener la configuración actual de bloqueo de objetos con los SDK de AWS, consulte, Uso de `GetObjectLockConfiguration` con un AWS SDK o la CLI.

Para ver un escenario interactivo que muestre diferentes características de bloqueo de objetos mediante AWS SDK, consulte Trabajo con las características de bloqueo de objetos de Amazon S3 mediante un SDK de AWS.

Para obtener información general sobre el uso de diferentes SDK de AWS, consulte Desarrollo con Amazon S3 mediante los SDK de AWS.

Establecer o modificar una retención legal en un objeto de S3

Puede establecer o eliminar una retención legal en un objeto de S3 mediante la consola de Amazon S3, AWS CLI, los SDK de AWS o la API de REST de Amazon S3.

> **Important**
>
> - Si desea establecer una retención legal en un objeto, el bucket del objeto debe tener ya activado el bloqueo de objetos.
> - Cuando se aplica PUT a la versión de un objeto que tiene un modo y un periodo de retención individual explícitos en un bucket, la configuración de bloqueo de objetos individual de la versión del objeto anula cualquier configuración de retención para las propiedades del bucket.

Para obtener más información, consulte the section called “Retenciones legales”.

Uso de la consola de S3

1. Inicie sesión en la AWS Management Console y abra la consola de Amazon S3 en https://console.aws.amazon.com/s3/.
2. En el panel de navegación izquierdo, elija Instancias.
3. En la lista Buckets, seleccione el nombre del bucket que contiene el objeto sobre el que desee establecer o modificar una retención legal.
4. En la lista Objetos, seleccione el objeto sobre el que desee establecer o modificar una retención legal.
5. En la página Propiedades del objeto, busque la sección Retención legal de bloqueo de objetos y seleccione Editar.
6. Seleccione Habilitar para establecer una retención legal o Deshabilitar para eliminar una retención legal.
7. Elija Guardar cambios.

Uso de la AWS CLI

En el siguiente ejemplo `put-object-legal-hold` se establece una retención legal sobre el objeto *`my-image.fs`* en el bucket denominado *`amzn-s3-demo-bucket1`*:

```
aws s3api put-object-legal-hold --bucket amzn-s3-demo-bucket1 --key my-image.fs --
legal-hold="Status=ON"
```

En el siguiente ejemplo `put-object-legal-hold` se elimina una retención legal sobre el objeto *`my-image.fs`* en el bucket denominado *`amzn-s3-demo-bucket1`*:

```
aws s3api put-object-legal-hold --bucket amzn-s3-demo-bucket1 --key my-image.fs --
legal-hold="Status=OFF"
```

Para obtener más información y ejemplos, consulte put-object-legal-hold en la referencia de comandos de AWS CLI.

> **Note**
>
> Puede ejecutar comandos de AWS CLI desde la consola mediante AWS CloudShell. AWS CloudShell es un intérprete de comandos previamente autenticado y basado en el navegador que se puede lanzar directamente desde la página web de la AWS Management Console.

Para obtener más información, consulte ¿Qué es CloudShell? en la Guía del usuario de AWS CloudShell.

Uso de la API de REST

Puede utilizar la API de REST para establecer o modificar una retención legal en un objeto. Para obtener más información, consulte PutObjectLegalHold en la Referencia de la API de Amazon Simple Storage Service.

Uso de los AWS SDK

Para ver ejemplos de cómo configurar una retención legal en un objeto con los SDK de AWS, consulte Uso de `PutObjectLegalHold` con un AWS SDK o la CLI.

Para ver ejemplos de cómo obtener el estado actual de retención legal con los SDK de AWS, consulte Obtención de la configuración de retención legal de un objeto de Amazon S3 mediante un SDK de AWS.

Para ver un escenario interactivo que muestre diferentes características de bloqueo de objetos mediante AWS SDK, consulte Trabajo con las características de bloqueo de objetos de Amazon S3 mediante un SDK de AWS.

Para obtener información general sobre el uso de diferentes SDK de AWS, consulte Desarrollo con Amazon S3 mediante los SDK de AWS.

Establecer o modificar un período de retención en un objeto de S3

Puede establecer o modificar un período de retención en un objeto de S3 mediante la consola de Amazon S3, AWS CLI, los SDK de AWS o la API de REST de Amazon S3.

Important

- Si desea establecer un período de retención en un objeto, el bucket del objeto debe tener ya activado el bloqueo de objetos.
- Cuando se aplica PUT a la versión de un objeto que tiene un modo y un periodo de retención individual explícitos en un bucket, la configuración de bloqueo de objetos individual de la versión del objeto anula cualquier configuración de retención para las propiedades del bucket.

- La única forma de eliminar un objeto en el modo de cumplimiento antes de que venza su fecha de retención es eliminar la Cuenta de AWS asociada.

Para obtener más información, consulte Periodos de retención.

Uso de la consola de S3

1. Inicie sesión en la AWS Management Console y abra la consola de Amazon S3 en https://console.aws.amazon.com/s3/.
2. En el panel de navegación izquierdo, elija Instancias.
3. En la lista Buckets, seleccione el nombre del bucket que contiene el objeto sobre el que desee establecer o modificar un período de retención.
4. En la lista Objetos, seleccione el objeto sobre el que desee establecer o modificar un período de retención.
5. En la página Propiedades del objeto, busque la sección Retención de bloqueo de objetos y seleccione Editar.
6. En Retención, seleccione Habilitar para establecer un período de retención o Deshabilitar para eliminar un período de retención.
7. Si se ha seleccionado Habilitar, en Modo de retención, elija el modo Gobierno o el modo Cumplimiento. Para obtener más información, consulte Modos de retención.
8. En Fecha límite de retención, elija la fecha en la que desea que finalice el período de retención. Durante este periodo, el objeto tiene protección WORM y no es posible sobrescribirlo ni eliminarlo. Para obtener más información, consulte Periodos de retención.
9. Elija Save changes (Guardar cambios).

Uso de la AWS CLI

En el siguiente ejemplo `put-object-retention` se establece un período de retención sobre el objeto *`my-image.fs`* en el bucket denominado *`amzn-s3-demo-bucket1`* hasta el 1 de enero de 2025:

```
aws s3api put-object-retention --bucket amzn-s3-demo-bucket1 --key my-image.fs --
retention='{ "Mode": "GOVERNANCE", "RetainUntilDate": "2025-01-01T00:00:00" }'
```

Para obtener más información y ejemplos, consulte put-object-retention en la referencia de comandos de AWS CLI.

> **Note**
>
> Puede ejecutar comandos de AWS CLI desde la consola mediante AWS CloudShell. AWS CloudShell es un intérprete de comandos previamente autenticado y basado en el navegador que se puede lanzar directamente desde la página web de la AWS Management Console. Para obtener más información, consulte ¿Qué es CloudShell? en la Guía del usuario de AWS CloudShell.

Uso de la API de REST

Puede utilizar la API de REST para establecer un periodo de retención en un objeto. Para obtener más información, consulte PutObjectRetention en la Referencia de la API de Amazon Simple Storage Service.

Uso de los AWS SDK

Para ver ejemplos de cómo configurar un período de retención en un objeto con los SDK de AWS, consulte Uso de `PutObjectRetention` con un AWS SDK o la CLI.

Para ver ejemplos de cómo obtener el período de retención en un objeto con los SDK de AWS, consulte Uso de `GetObjectRetention` con un AWS SDK o la CLI.

Para ver un escenario interactivo que muestre diferentes características de bloqueo de objetos mediante AWS SDK, consulte Trabajo con las características de bloqueo de objetos de Amazon S3 mediante un SDK de AWS.

Para obtener información general sobre el uso de diferentes SDK de AWS, consulte Desarrollo con Amazon S3 mediante los SDK de AWS.

Establecer o modificar un período de retención predeterminado en un bucket de S3

Puede establecer o modificar un período de retención predeterminado en un bucket de S3 mediante la consola de Amazon S3, AWS CLI, los SDK de AWS o la API de REST de Amazon S3. Se especifica una duración, en días o años, para proteger cada versión del objeto colocada en el bucket.

Important

- Si desea establecer un periodo de retención predeterminado en un bucket, el bucket debe contar ya con Bloqueo de objetos activado.
- Cuando se aplica PUT a la versión de un objeto que tiene un modo y un periodo de retención individual explícitos en un bucket, la configuración de bloqueo de objetos individual de la versión del objeto anula cualquier configuración de retención para las propiedades del bucket.
- La única forma de eliminar un objeto en el modo de cumplimiento antes de que venza su fecha de retención es eliminar la Cuenta de AWS asociada.

Para obtener más información, consulte Periodos de retención.

Uso de la consola de S3

1. Inicie sesión en la AWS Management Console y abra la consola de Amazon S3 en https://console.aws.amazon.com/s3/.
2. En el panel de navegación izquierdo, elija Instancias.
3. En la lista Buckets, seleccione el nombre del bucket sobre el que desea establecer o modificar un período de retención predeterminado.
4. Elija la pestaña Propiedades.
5. En Propiedades, desplácese hacia abajo hasta la sección Bloqueo de objetos y seleccione Editar.
6. En Retención predeterminada, seleccione Habilitar para establecer un período de retención o Deshabilitar para eliminar un período de retención.
7. Si se ha seleccionado Habilitar, en Modo de retención, elija el modo Gobierno o el modo Cumplimiento. Para obtener más información, consulte Modos de retención.
8. En Período de retención predeterminado, seleccione el número de días o años que desea que dure el período de retención. Los objetos que se coloquen en este bucket se bloquearán durante ese número de días o años. Para obtener más información, consulte Periodos de retención.
9. Elija Save changes (Guardar cambios).

Uso de la AWS CLI

El siguiente comando de ejemplo `put-object-lock-configuration` establece un período de retención de Bloqueo de objetos de 50 días en un bucket denominado *amzn-s3-demo-bucket1* mediante el modo de cumplimiento:

```
aws s3api put-object-lock-configuration --bucket amzn-s3-demo-bucket1 --object-
lock-configuration='{ "ObjectLockEnabled": "Enabled", "Rule": { "DefaultRetention":
 { "Mode": "COMPLIANCE", "Days": 50 }}}'
```

En el siguiente ejemplo `put-object-lock-configuration` se elimina la configuración de retención predeterminada de un bucket:

```
aws s3api put-object-lock-configuration --bucket amzn-s3-demo-bucket1 --object-lock-
configuration='{ "ObjectLockEnabled": "Enabled"}'
```

Para obtener más información y ejemplos, consulte put-object-lock-configuration en la referencia de comandos de AWS CLI.

> **Note**
>
> Puede ejecutar comandos de AWS CLI desde la consola mediante AWS CloudShell. AWS CloudShell es un intérprete de comandos previamente autenticado y basado en el navegador que se puede lanzar directamente desde la página web de la AWS Management Console. Para obtener más información, consulte ¿Qué es CloudShell? en la Guía del usuario de AWS CloudShell.

Uso de la API de REST

Puede utilizar la API de REST para establecer un periodo de retención predeterminado en un bucket de S3 existente. Para obtener más información, consulte PutObjectLockConfiguration en la Referencia de la API de Amazon Simple Storage Service.

Uso de los AWS SDK

Para ver ejemplos de cómo establecer un período de retención predeterminado en un bucket de S3 existente con los SDK de AWS, consulte Uso de `PutObjectLockConfiguration` con un AWS SDK o la CLI.

Para ver un escenario interactivo que muestre diferentes características de bloqueo de objetos mediante AWS SDK, consulte Trabajo con las características de bloqueo de objetos de Amazon S3 mediante un SDK de AWS.

Para obtener información general sobre el uso de diferentes SDK de AWS, consulte Desarrollo con Amazon S3 mediante los SDK de AWS.

Uso de las clases de almacenamiento de Amazon S3

Cada objeto de Amazon S3 tiene una clase de almacenamiento asociada. Por ejemplo, si enumera los objetos de un bucket de S3, la consola muestra la clase de almacenamiento de todos los objetos en la lista. Amazon S3 ofrece una gama de clases de almacenamiento para los objetos que almacene. Debe seleccionar una clase de almacenamiento en función de su escenario de caso de uso y sus requisitos de acceso y rendimiento. Todas estas clases de almacenamiento ofrecen una alta durabilidad.

Las siguientes secciones proporcionan información sobre las diferentes clases de almacenamiento y cómo establecer las clases de almacenamiento para los objetos.

Temas

Clases de almacenamiento para objetos a los que se obtiene acceso con frecuencia

Para casos de uso sensibles al desempeño (aquellos que necesitan un tiempo de acceso de milisegundos) y datos a los que se obtiene acceso frecuentemente, Amazon S3 proporciona las siguientes clases de almacenamiento:

- S3 Standard: la clase de almacenamiento predeterminada. Si no se especifica una clase de almacenamiento al cargar un objeto, Amazon S3 asigna la clase de almacenamiento S3 Standard.
- S3 Express One Zone: Amazon S3 Express One Zone es una clase de almacenamiento de Amazon S3 en zona única de alto rendimiento que está diseñada específicamente para ofrecer acceso constante a los datos en milisegundos de un solo dígito para los datos a los que accede para las aplicaciones sensibles a la latencia. S3 Express One Zone es la clase de almacenamiento de objetos en la nube con la latencia más baja disponible en la actualidad, con una velocidad de acceso a los datos hasta 10 veces más rápida y unos costos de solicitud un 50 % más bajos que los de S3 Standard. Con S3 Express One Zone, sus datos se almacenan de forma redundante en varios dispositivos dentro de una única zona de disponibilidad. Para obtener más información, consulte ¿Qué es S3 Express One Zone?.
- Redundancia reducida: la clase de almacenamiento de redundancia reducida (RRS) se ha diseñado para los datos no críticos y reproducibles que se pueden almacenar con menor redundancia que la clase de almacenamiento S3 Standard.

  > **Important**
  >
  > Recomendamos no usar esta clase de almacenamiento. La clase de almacenamiento S3 Standard es más económica.

  En cuanto a durabilidad, los objetos RRS tienen una pérdida anual esperada media del 0,01 % de los objetos. Si se pierde un objeto RRS, Amazon S3 devuelve un error 405 cuando se realizan solicitudes de ese objeto.

Clase de almacenamiento para optimizar automáticamente los datos con patrones de acceso cambiantes o desconocidos

S3 Intelligent-Tiering es una clase de almacenamiento de Amazon S3 diseñada para optimizar los costos de almacenamiento mediante el traslado automático de los datos a la capa de acceso más rentable sin que se produzca un impacto en el rendimiento o una sobrecarga operativa. S3 Intelligent-Tiering es la única clase de almacenamiento en la nube que ofrece un ahorro automático en los costos mediante la migración de datos en un nivel de objeto detallado entre capas de acceso cuando los patrones de acceso cambian. S3 Intelligent-Tiering es la clase de almacenamiento ideal cuando desea optimizar los costos de almacenamiento de datos con patrones de acceso desconocidos o cambiantes. S3 Intelligent-Tiering no tiene tarifas de recuperación.

Por una pequeña tarifa mensual de monitoreo y automatización de objetos, S3 Intelligent-Tiering monitorea los patrones de acceso y traslada automáticamente los objetos a los que no se ha accedido para reducir los niveles de acceso de costos. S3 Intelligent-Tiering ofrece ahorros automáticos en los costes de almacenamiento en tres niveles de acceso de baja latencia y alto rendimiento. Para los datos a los que se puede acceder de forma asíncrona, puede optar por activar las capacidades de archivo automático dentro de la clase de almacenamiento de S3 Intelligent-Tiering. S3 Intelligent-Tiering está diseñado para ofrecer una disponibilidad del 99.9 % y una durabilidad del 99.9999999 %.

S3 Intelligent-Tiering almacena automáticamente objetos en tres niveles de acceso:

- Acceso frecuente: los objetos cargados en S3 Intelligent-Tiering o migrados allí se almacenan de forma automática en la capa de acceso frecuente.
- Acceso poco frecuente: S3 Intelligent-Tiering traslada a la capa de acceso poco frecuente aquellos objetos a los que no se accedió en 30 días consecutivos.
- Acceso instantáneo a archivos: con S3 Intelligent-Tiering, cualquier objeto existente al que no se haya accedido durante 90 días consecutivos pasará automáticamente a la capa de acceso instantáneo a archivos.

Además de estos tres niveles, S3 Intelligent-Tiering ofrece dos niveles opcionales de acceso al archivo:

- Acceso de archivo: S3 Intelligent-Tiering le proporciona la opción de activar el nivel acceso de archivo para datos a los que se puede acceder de forma asíncrona. Después de la activación, el nivel Archive Access archiva automáticamente objetos a los que no se accedió durante un mínimo de 90 días consecutivos.
- Acceso de archivo profundo: S3 Intelligent-Tiering le proporciona la opción de activar el nivel acceso de archivo profundo para datos a los que se puede acceder de forma asíncrona. Después de la activación, el nivel Deep Archive Access archiva automáticamente objetos a los que no se accedió durante un mínimo de 180 días consecutivos.

> **Note**
>
> - Active el nivel Archive Access solo durante 90 días si desea omitir el nivel Archive Instant Access. El nivel Archive Access ofrece un almacenamiento de costo ligeramente

inferior con tiempos de recuperación de minutos a horas. El nivel Archive Instant Access proporciona acceso en milisegundos y un alto rendimiento.

- Active las capas de acceso a archivo y archivo profundo solo si la aplicación puede acceder de forma asíncrona a sus objetos. Si el objeto que está recuperando está almacenado en las capas de acceso a archivo o archivo profundo, primero restaure el objeto mediante `RestoreObject`.

Puede desplazar los datos recién creados a S3 Intelligent-Tiering al definirlo como la clase de almacenamiento predeterminada. También puede activar uno o ambos niveles de acceso a archivos mediante la operación de la API `PutBucketIntelligentTieringConfiguration`, la AWS CLI o la consola de Amazon S3. Para obtener información acerca del uso de S3 Intelligent-Tiering y la activación de los niveles de acceso al archivo, consulte Uso de S3 Intelligent-Tiering.

Para acceder a los objetos de los niveles Acceso a archivos o Acceso a archivos profundo, primero debe restaurarlos. Para obtener más información, consulte Restauración de objetos desde los niveles S3 Intelligent-Tiering Archive Access o Deep Archive Access.

> **Note**
>
> Si el tamaño del objeto es inferior a 128 KB, no se monitorea y no es elegible para las capas automáticas. Los objetos más pequeños siempre se almacenan en el nivel Acceso frecuente. Para obtener más información acerca de S3 Intelligent-Tiering, consulte Capas de acceso de S3 Intelligent-Tiering.

Clases de almacenamiento para objetos a los que se obtiene acceso con poca frecuencia

Las clases de almacenamiento S3 Standard-IA y S3 One Zone-IA están diseñadas para datos de larga duración y acceso poco frecuente. (IA significa acceso poco frecuente, infrequent access). Los objetos S3 Standard-IA y S3 One Zone-IA están disponibles para acceder en milisegundos (de modo similar a la clase de almacenamiento S3 Standard). Amazon S3 cobra una tarifa de recuperación para estos objetos, por lo que son más adecuados para los datos a los que se obtiene acceso con poca frecuencia. Para obtener información acerca de los precios, consulte Precios de Amazon S3.

Por ejemplo, puede elegir que las clases S3 Standard-IA y S3 One Zone-IA hagan lo siguiente:

- Para almacenar copias de seguridad.
- Para datos más antiguos a los que se obtiene acceso con poca frecuencia, pero que todavía necesitan acceso en milisegundos. Por ejemplo, cuando cargue datos, puede elegir la clase de almacenamiento S3 Standard y utilizar la configuración del ciclo de vida para indicar a Amazon S3 que realice la transición de los objetos a la clase S3 Standard-IA o S3 One Zone-IA.

  Para obtener más información sobre la administración del ciclo de vida, consulte Administración del ciclo de vida del almacenamiento.

> **Note**
>
> Las clases de almacenamiento S3 Standard-IA y S3 One Zone-IA son adecuados para los objetos de más de 128 KB que se desean almacenar durante al menos 30 días. Si un objeto tiene menos de 128 KB, Amazon S3 cobra por 128 KB. Si se elimina un objeto antes de que termine el periodo mínimo de almacenamiento de 30 días, se cobrará por 30 días. A los objetos que se eliminen, sobrescriban o pasen a una clase de almacenamiento diferente antes de 30 días se les aplicará el cargo por uso de almacenamiento normal más un cargo prorrateado durante el resto del mínimo de 30 días. Para obtener información acerca de los precios, consulte Precios de Amazon S3.

Estas clases de almacenamiento se diferencian en lo siguiente:

- S3 Standard-IA: Amazon S3 almacena los datos de los objetos de forma redundante en varias zonas de disponibilidad separadas geográficamente (de forma similar a la clase de almacenamiento S3 Standard). Los objetos S3 Standard-IA resisten a la pérdida de una zona de disponibilidad. Esta clase de almacenamiento ofrece mayor disponibilidad y resistencia que la clase S3 One Zone-IA.
- S3 One Zone-IA: Amazon S3 almacena los datos de los objetos en una sola zona de disponibilidad, lo que resulta más económico que la clase S3 Standard-IA. Sin embargo, los datos no son resistentes a la pérdida física de la zona de disponibilidad como consecuencia de desastres, como terremotos e inundaciones. La clase de almacenamiento S3 One Zone-IA es tan duradera como la clase S3 Standard-IA, pero tiene menor disponibilidad y resistencia. Para ver una comparación de durabilidad y disponibilidad entre las distintas clases de almacenamiento, consulte Comparación de las clases de almacenamiento de Amazon S3 al final de esta sección. Para obtener información acerca de los precios, consulte Precios de Amazon S3.

Le recomendamos lo siguiente:

- S3 Standard-IA: utilice esta clase para la copia principal o única de los datos que no se puedan volver a crear.
- S3 One Zone-IA: utilice esta clase si puede volver a crear los datos cuando la zona de disponibilidad produce un error y para las réplicas de objetos cuando se configura la replicación entre regiones de S3 (CRR).

Clases de almacenamiento para objetos a los que se accede con poca frecuencia

Las clases de almacenamiento S3 Glacier Instant Retrieval, S3 Glacier Flexible Retrieval y S3 Glacier Deep Archive están diseñadas para el archivo de datos y el almacenamiento de datos de forma económica y a largo plazo. Estas clases de almacenamiento ofrecen la misma durabilidad y resistencia que las clases de almacenamiento S3 Standard y S3 Standard-IA. Para obtener más información acerca de las clases de almacenamiento de S3 Glacier, consulte Almacenamiento de datos a largo plazo con clases de almacenamiento de S3 Glacier.

Amazon S3 ofrece las siguientes clases de almacenamiento S3 Glacier:

- S3 Glacier Instant Retrieval: se utiliza para datos a largo plazo a los que se accede con poca frecuencia y requieren recuperación en milisegundos. Los datos de esta clase de almacenamiento están disponibles para su acceso en tiempo real.
- S3 Glacier Flexible Retrieval: se utiliza para archivos en los que puede ser necesario recuperar partes de los datos en cuestión de minutos. Los datos de esta clase de almacenamiento se archivan y no están disponibles para su acceso en tiempo real.
- S3 Glacier Deep Archive: se usa para archivar datos a los que se necesita obtener acceso en contadas ocasiones. Los datos de esta clase de almacenamiento se archivan y no están disponibles para su acceso en tiempo real.

Recuperación de objetos archivados

Puede establecer la clase de almacenamiento de un objeto en S3 Glacier Flexible Retrieval o S3 Glacier Deep Archive del mismo modo que lo hace para otras clases de almacenamiento como se describe en la sección Establecimiento de la clase de almacenamiento de un objeto. Sin embargo, los objetos de S3 Glacier Flexible Retrieval y S3 Glacier Deep Archive se archivan y no están

disponibles para su acceso en tiempo real. Para obtener más información, consulte Almacenamiento de archivos.

> **Note**
>
> Cuando utiliza las clases de almacenamiento de S3 Glacier, sus objetos permanecen en Amazon S3. No puede acceder a ellos directamente a través del servicio independiente de Amazon S3 Glacier. Para obtener más información sobre el servicio de Amazon S3 Glacier, consulte la Guía para desarrolladores de Amazon S3 Glacier.

Clase de almacenamiento para Amazon S3 en Outposts

Con Amazon S3 en Outposts, puede crear buckets de S3 en recursos de AWS Outposts y almacenar y recuperar objetos en las instalaciones para las aplicaciones que requieren acceso local a los datos, procesamiento local de los datos y residencia de los datos. Puede usar las mismas operaciones y características de la API en AWS Outposts que en Amazon S3, incluidas las políticas de acceso, el cifrado y el etiquetado. Puede utilizar S3 en Outposts a través de la AWS Management Console, la AWS CLI, los SDK de AWS o la API de REST.

S3 en Outposts proporciona una nueva clase de almacenamiento, S3 Outposts (`OUTPOSTS`). La clase de almacenamiento S3 Outposts está disponible solo para los objetos almacenados en buckets que se encuentran en Outposts. Si intenta utilizar esta clase de almacenamiento con un bucket de S3 en una Región de AWS, se producirá un error `InvalidStorageClass`. Además, si intenta usar otras clases de almacenamiento de S3 con objetos almacenados en buckets de S3 en Outposts, se producirá el mismo error.

Los objetos almacenados en la clase de almacenamiento S3 Outposts (`OUTPOSTS`) siempre se cifran mediante cifrado del lado del servidor con claves de cifrado administradas (SSE-S3) de Amazon S3. Para obtener más información, consulte Uso del cifrado del servidor con claves administradas por Amazon S3 (SSE-S3).

También puede elegir explícitamente cifrar objetos almacenados en la clase de almacenamiento S3 Outposts mediante cifrado del lado del servidor con claves de cifrado del cliente (SSE-C). Para obtener más información, consulte Uso de cifrado en el lado del servidor con claves proporcionadas por el cliente (SSE-C).

> **Note**
>
> S3 en Outposts no es compatible con el cifrado del lado del servidor con claves AWS Key Management Service (AWS KMS) (SSE-KMS).

Para obtener más información sobre S3 en Outposts, consulte ¿Qué es Amazon S3 en Outposts?

Comparación de las clases de almacenamiento de Amazon S3

En la siguiente tabla se comparan las clases de almacenamiento, incluidas su disponibilidad, durabilidad, duración mínima del almacenamiento y otras cuestiones.

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 requiere 40 KB de metadatos adicionales para cada objeto archivado. Esto incluye 32 KB de metadatos cobrados a la velocidad de recuperación flexible de S3 Glacier (necesaria para identificar y recuperar los datos) y 8 KB de datos adicionales cobrados a la tarifa estándar de S3. La tarifa estándar de S3 es necesaria para mantener el nombre y los metadatos definidos por el usuario para los objetos archivados en S3 Glacier Flexible Retrieval. Para obtener más información acerca de las clases de almacenamiento, consulte Clases de almacenamiento de Amazon S3.

** S3 Glacier Deep Archive requiere 40 KB de metadatos adicionales para cada objeto archivado. Esto incluye 32 KB de metadatos cobrados a la tarifa S3 Glacier Deep Archive (necesaria para identificar y recuperar los datos) y 8 KB de datos adicionales cobrados a la tarifa estándar de S3. La

tarifa estándar de S3 es necesaria para mantener el nombre y los metadatos definidos por el usuario para los objetos archivados en Amazon S3 Glacier Deep Archive. Para obtener más información acerca de las clases de almacenamiento, consulte Clases de almacenamiento de Amazon S3.

Tenga en cuenta que todas las clases de almacenamiento, excepto S3 One Zone-IA y S3 Express One Zone, están diseñadas para resistir la pérdida física de una zona de disponibilidad como consecuencia de desastres. Además de los requisitos de desempeño del escenario de su aplicación, tenga en cuenta los costos. Para obtener información sobre el precio de las clases de almacenamiento, consulte Precios de Amazon S3.

Establecimiento de la clase de almacenamiento de un objeto

Para establecer y actualizar las clases de almacenamiento de objetos, puede utilizar la consola de Amazon S3, los SDK de AWS o la AWS Command Line Interface (AWS CLI). Todos estos enfoques utilizan las operaciones de la API de Amazon S3 para enviar solicitudes a Amazon S3.

Las operaciones de la API de Amazon S3 permiten configurar (o actualizar) la clase de almacenamiento de los objetos, tal como se indica a continuación:

- Al crear un objeto, puede especificar su clase de almacenamiento. Por ejemplo, al crear objetos utilizando las operaciones de API PUT Object, POST Object e Initiate Multipart Upload (Iniciar carga multiparte), se debe agregar el encabezado de solicitud `x-amz-storage-class` para especificar una clase de almacenamiento. Si no se agrega este encabezado, Amazon S3 utiliza S3 Standard, la clase de almacenamiento predeterminada.
- También puede cambiar la clase de almacenamiento de un objeto que ya está almacenado en Amazon S3 a cualquier otra clase de almacenamiento haciendo una copia del objeto mediante la operación de la API PUT Object - Copy. Sin embargo, no puede utilizar PUT Object - Copy para copiar objetos que están almacenados en las clases S3 Glacier Flexible Retrieval o S3 Glacier Deep Archive. Tampoco se puede realizar la transición de S3 One Zone-IA a S3 Glacier Instant Retrieval.

  Debe copiar el objeto en el mismo bucket utilizando el mismo nombre de clave y especificando encabezados de solicitud como se indica a continuación:

  - Establezca el encabezado `x-amz-metadata-directive` en `COPY`.
  - Establezca el encabezado `x-amz-storage-class` en la clase de almacenamiento que desea utilizar.

En un bucket con control de versiones habilitado, no puede modificar la clase de almacenamiento de una versión específica de un objeto. Al copiar el objeto, Amazon S3 le da un nuevo ID de versión.

- Puede cambiar la clase de almacenamiento de un objeto mediante la consola de Amazon S3 si el tamaño del objeto es inferior a 160 GB. Si es mayor, se recomienda añadir la configuración del ciclo de vida de S3 para cambiar la clase de almacenamiento del objeto.
- Si utiliza la consola de Amazon S3 para cambiar la clase de almacenamiento de un objeto que tiene etiquetas definidas por el usuario, debe tener el permiso `s3:GetObjectTagging`. Si va a cambiar la clase de almacenamiento de un objeto que no tiene etiquetas definidas por el usuario pero que tiene un tamaño superior a 16 MB, también debe tener el permiso `s3:GetObjectTagging`. Si la política de bucket de destino deniega la acción `s3:GetObjectTagging`, la clase de almacenamiento del objeto se actualizará, pero las etiquetas definidas por el usuario se eliminarán del objeto y aparecerá un error.
- Puede indicar a Amazon S3 que modifique la clase de almacenamiento de los objetos añadiendo la configuración del ciclo de vida de S3 a un bucket. Para obtener más información, consulte Administración del ciclo de vida del almacenamiento.
- Cuando establezca una configuración de replicación, puede especificar la clase de almacenamiento de los objetos replicados como cualquier otra clase de almacenamiento. Sin embargo, no puede replicar objetos que están almacenados en las clases S3 Glacier Flexible Retrieval o S3 Glacier Deep Archive. Para obtener más información, consulte Configuración de replicación.

Restringir permisos de política de acceso a una clase de almacenamiento específica

Cuando concede permisos de política de acceso para operaciones de Amazon S3, puede usar la clave de condición `s3:x-amz-storage-class` para restringir la clase de almacenamiento que se debe utilizar al almacenar objetos cargados. Por ejemplo, cuando concede el permiso de `s3:PutObject`, puede restringir las cargas de objetos a una clase de almacenamiento específica. Para ver una política de ejemplo, consulte Ejemplo: restricción de cargas de objetos a objetos con una clase de almacenamiento específica.

Para obtener más información sobre el uso de condiciones en políticas y una lista completa de claves de condición de Amazon S3, consulte los temas siguientes:

- Actions, resources, and condition keys for Amazon S3 en la Referencia de autorizaciones de servicio

- Ejemplos de políticas de bucket que utilizan claves de condición

Almacenamiento de datos a largo plazo con clases de almacenamiento de S3 Glacier

Amazon S3 ofrece varias clases de almacenamiento S3 Glacier diseñadas para ofrecer soluciones rentables para almacenar datos a largo plazo a los que no se accede con frecuencia. Las clases de almacenamiento de S3 Glacier son las siguientes:

- S3 Glacier Instant Retrieval
- S3 Glacier Flexible Retrieval
- S3 Glacier Deep Archive

Usted elige una de estas clases de almacenamiento en función de la frecuencia con la que accede a sus datos y de la rapidez con la que necesita recuperarlos. Cada una de estas clases de almacenamiento ofrece la misma durabilidad y resiliencia que la clase de almacenamiento S3 Standard, pero con costos de almacenamiento más bajos. Para obtener más información acerca de las clases de almacenamiento de S3 Glacier, consulte https://aws.amazon.com/s3/storage-classes/glacier/.

Temas

- Comparación de las clases de almacenamiento de S3 Glacier
- S3 Glacier Instant Retrieval
- S3 Glacier Flexible Retrieval
- S3 Glacier Deep Archive
- Almacenamiento de archivos
- En qué se diferencian estas clases de almacenamiento del servicio S3 Glacier

Comparación de las clases de almacenamiento de S3 Glacier

Cada clase de almacenamiento de S3 Glacier tiene una duración mínima de almacenamiento para todos los objetos. Si ha eliminado, sobrescrito o movido el objeto a una clase de almacenamiento diferente antes del periodo mínimo, se le cobrará el periodo completo de almacenamiento mínimo.

Algunas clases de almacenamiento de S3 Glacier son de archivo, lo que significa que los objetos almacenados en esas clases están archivados y no están disponibles para su acceso en tiempo real. Para obtener más información, consulte Almacenamiento de archivos.

Las clases de almacenamiento diseñadas para patrones de acceso menos frecuentes con tiempos de recuperación más largos ofrecen costos de almacenamiento más bajos. Para obtener información sobre precios, consulte https://aws.amazon.com/s3/pricing/.

En la siguiente tabla se resumen los puntos clave que se deben tener en cuenta al elegir una clase de almacenamiento de S3 Glacier:

S3 Glacier Instant Retrieval

Recomendamos utilizar S3 Glacier Instant Retrieval para los datos a largo plazo a los que se accede una vez por trimestre y que requieren tiempos de recuperación de milisegundos. Esta clase de almacenamiento es ideal para casos de uso sensibles al rendimiento, como el alojamiento de imágenes, las aplicaciones de uso compartido de archivos y el almacenamiento de registros médicos para poder acceder a ellos durante consultas.

La clase de almacenamiento S3 Glacier Instant Retrieval ofrece acceso en tiempo real a sus objetos con la misma latencia y rendimiento que la clase de almacenamiento S3 Standard-IA. Si se compara con S3 Standard-IA, S3 Glacier Instant Retrieval ofrece menores costos de almacenamiento, pero costos de acceso de datos más altos.

Hay un tamaño mínimo de objeto de 128 KB para los datos almacenados en la clase de almacenamiento S3 Glacier Instant Retrieval. Esta clase de almacenamiento también tiene un periodo mínimo de almacenamiento de 90 días.

S3 Glacier Flexible Retrieval

Recomendamos utilizar S3 Glacier Flexible Retrieval para archivar los datos a los que se accede una o dos veces al año y que no requieren acceso inmediato. S3 Glacier Flexible Retrieval ofrece tiempos de recuperación flexibles para ayudarle a equilibrar los costos, con tiempos de acceso que van desde unos minutos hasta horas y recuperaciones masivas gratuitas. Esta clase de almacenamiento es ideal para las copias de seguridad y la recuperación de desastres.

Los objetos almacenados en S3 Glacier Flexible Retrieval se archivan y no están disponibles para su acceso en tiempo real. Para obtener más información, consulte Almacenamiento de archivos. Para obtener acceso a estos objetos, primero debe iniciar una solicitud de restauración que crea una copia temporal del objeto a la que puede acceder cuando se complete la solicitud. Para obtener más

información, consulte Trabajar con objetos archivados. Al restaurar un objeto, puede elegir un nivel de recuperación que se adapte a su caso de uso, con costos más bajos y tiempos de restauración más largos.

Los siguientes niveles de recuperación están disponibles para S3 Glacier Flexible Retrieval:

- Recuperación acelerada: normalmente, se restaura el objeto en 1 a 5 minutos. Las recuperaciones aceleradas están sujetas a demanda, por lo que, para asegurarse de que cuenta con tiempos de restauración fiables y predecibles, le recomendamos que adquiera capacidad de recuperación aprovisionada. Para obtener más información, consulte Capacidad aprovisionada.
- Recuperación estándar: normalmente restaura el objeto en 3 a 5 horas, o entre 1 minuto y 5 horas cuando se utiliza Operaciones por lotes de S3. Para obtener más información, consulte Restaurara objetos con Operaciones por lotes.
- Recuperación masiva: suele restaurarse el objeto en un plazo de entre 5 y 12 horas. Las recuperaciones masivas son gratuitas.

La duración mínima de almacenamiento de los objetos en la clase de almacenamiento S3 Glacier Flexible Retrieval es de 90 días.

S3 Glacier Flexible Retrieval requiere 40 KB de metadatos adicionales para cada objeto. Esto incluye 32 KB de metadatos necesarios para identificar y recuperar los datos, que se cobran a la tarifa predeterminada para S3 Glacier Flexible Retrieval. Se requieren 8 KB de datos adicionales para mantener el nombre y los metadatos definidos por el usuario para los objetos archivados y se cobra a la tarifa de S3 Standard.

S3 Glacier Deep Archive

Recomendamos usar S3 Glacier Deep Archive para archivar datos a los que se accede menos de una vez al año. Esta clase de almacenamiento está diseñada para conservar conjuntos de datos durante varios años, con el fin de cumplir con los requisitos de cumplimiento y también se puede utilizar para realizar copias de seguridad o recuperación de desastres, o bien para cualquier dato al que se acceda con poca frecuencia y que se pueda esperar hasta 72 horas para recuperarlo. S3 Glacier Deep Archive es la opción de almacenamiento más económica de AWS.

Los objetos almacenados en S3 Glacier Flexible Retrieval se archivan y no están disponibles para su acceso en tiempo real. Para obtener más información, consulte Almacenamiento de archivos. Para obtener acceso a estos objetos, primero debe iniciar una solicitud de restauración que crea una copia temporal del objeto a la que puede acceder cuando se complete la solicitud. Para obtener más

información, consulte Trabajar con objetos archivados. Al restaurar un objeto, puede elegir un nivel de recuperación que se adapte a su caso de uso, con costos más bajos y tiempos de restauración más largos.

Los siguientes niveles de recuperación están disponibles para S3 Glacier Deep Archive:

- Recuperación estándar: normalmente restaura el objeto en 12 horas o entre 9 y 12 horas cuando se utiliza Operaciones por lotes de S3. Para obtener más información, consulte Restaurara objetos con Operaciones por lotes.
- Recuperación masiva: normalmente, se restaura el objeto en un plazo de 48 horas a una fracción del costo del nivel de recuperación Standard.

La duración mínima de almacenamiento de los objetos en la clase de almacenamiento S3 Glacier Deep Archive es de 180 días.

S3 Glacier Deep Archive requiere 40 KB de metadatos adicionales para cada objeto archivado. Esto incluye 32 KB de metadatos necesarios para identificar y recuperar los datos, que se cobran a la tarifa predeterminada para S3 Glacier Deep Archive. Se requieren 8 KB de datos adicionales para mantener el nombre y los metadatos definidos por el usuario para los objetos archivados y se cobra a la tarifa de S3 Standard.

Almacenamiento de archivos

S3 Glacier Flexible Retrieval y S3 Glacier Deep Archive son clases de almacenamiento de archivo. Esto significa que cuando se almacena un objeto en estas clases de almacenamiento, ese objeto se archiva y no se puede acceder a él directamente. Para acceder a un objeto archivado, debe enviar una solicitud de restauración y esperar a que el servicio restaure el objeto. La solicitud de restauración restaura una copia temporal del objeto y esa copia se elimina cuando vence la duración especificada en la solicitud. Para obtener más información, consulte Trabajar con objetos archivados.

Estas clases de almacenamiento requieren 40 KB de metadatos adicionales para cada objeto archivado. Esto incluye 32 KB de metadatos necesarios para identificar y recuperar los datos, que se cobran a la tarifa predeterminada para esa clase de almacenamiento. Se requieren 8 KB de datos adicionales para mantener el nombre y los metadatos definidos por el usuario para los objetos archivados y se cobra a la tarifa de S3 Standard.

Los objetos de estas clases de almacenamiento se facturan según las tarifas de clase de almacenamiento de S3 Standard cuando los carga mediante cargas multiparte. Para obtener más información, consulte Carga multiparte y precios.

Puede restaurar objetos archivados en estas clases de almacenamiento con hasta 1000 transacciones por segundo (TPS) de solicitudes de restauración de objetos por cuenta y por Región de AWS.

En qué se diferencian estas clases de almacenamiento del servicio S3 Glacier

Las clases de almacenamiento de S3 Glacier forman parte del servicio Amazon S3 y almacenan datos como objetos en buckets de S3. Puede administrar objetos en estas clases de almacenamiento mediante la consola de S3 o mediante programación con las API o SDK de S3. Al almacenar objetos en las clases de almacenamiento de S3 Glacier, puede utilizar características de S3, como el cifrado avanzado, el etiquetado de objetos y las configuraciones de S3 Lifecycle, para ayudar a gestionar la accesibilidad y el costo de los datos.

> **Important**
>
> Recomendamos utilizar las clases de almacenamiento de S3 Glacier dentro del servicio Amazon S3 para todos sus datos a largo plazo.

El servicio Amazon S3 Glacier (S3 Glacier) es un servicio independiente que almacena los datos como archivos en almacenes. Este servicio no es compatible con las características de Amazon S3 y no proporciona soporte de consola para las operaciones de carga y descarga de datos. No recomendamos usar el servicio de S3 Glacier para sus datos a largo plazo. No se puede acceder a los datos almacenados en este servicio desde el servicio Amazon S3. Si desea obtener información sobre el servicio de S3 Glacier, consulte la Guía para desarrolladores de Amazon S3 Glacier. Para transferir datos del servicio Amazon S3 Glacier a una clase de almacenamiento en Amazon S3, consulte Transferencia de datos de almacenes de Amazon S3 Glacier a Amazon S3 en la biblioteca de soluciones de AWS.

Amazon S3 Intelligent Tiering

La clase de almacenamiento S3 Intelligent-Tiering se diseñó para optimizar los costos de almacenamiento mediante el desplazamiento automático de los datos al nivel de acceso de almacenamiento más rentable cuando cambian los patrones de acceso, sin que afecte al rendimiento ni se produzca sobrecarga operativa. Por un pequeño cargo mensual de monitoreo y automatización de objetos, S3 Intelligent-Tiering monitorea los patrones de acceso y traslada automáticamente los objetos de una capa a otra.

S3 Intelligent-Tiering ofrece ahorros automáticos en los costos de almacenamiento en tres niveles de acceso de baja latencia y alto rendimiento. Para los datos a los que se puede acceder de forma asíncrona, puede optar por activar las capacidades de archivo automático dentro de la clase de almacenamiento de S3 Intelligent-Tiering. S3 Intelligent-Tiering no tiene cargos de recuperación. Si se obtiene acceso a un objeto del nivel Infrequent Access o Archive Instant Access, este se desplaza automáticamente al nivel Infrequent Access. No se aplican cargos adicionales a las capas cuando los objetos se mueven entre las capas dentro del tipo de almacenamiento S3 Intelligent-Tiering.

S3 Intelligent-Tiering es la clase de almacenamiento recomendada para datos con patrones de acceso desconocidos, cambiantes o impredecibles, independientemente del tamaño del objeto o del período de retención, como lagos de datos, análisis de datos y nuevas aplicaciones.

Para obtener información acerca del uso de S3 Intelligent-Tiering, consulte las siguientes secciones:

Temas

Cómo funciona S3 Intelligent-Tiering

La clase de almacenamiento Amazon S3 Intelligent-Tiering almacena automáticamente los objetos en tres niveles de acceso. Un nivel está optimizado para el acceso frecuente, un nivel de menor costo está optimizado para un acceso poco frecuente y otro nivel de muy bajo costo está optimizado para datos a los que se accede con muy poca frecuencia. Por un módico precio mensual de monitorización y automatización de objetos, S3 Intelligent-Tiering monitoriza los patrones de acceso y mueve automáticamente los objetos al nivel Infrequent Access cuando no se ha accedido a ellos durante 30 días consecutivos. Después de 90 días sin acceso, los objetos se mueven al nivel Archive Instant Access sin impacto en el rendimiento ni sobrecarga operativa.

Para obtener el costo de almacenamiento más bajo para datos a los que se puede acceder en cuestión de minutos u horas, active las capacidades de archivo para agregar dos niveles de acceso adicionales. Puede pasar objetos a la capa de acceso de archivo, la capa de acceso de archivo profundo o ambas. Con Archive Access, S3 Intelligent-Tiering mueve los objetos a los que no se ha accedido durante un mínimo de 90 días consecutivos al nivel Archive Access. Con Acceso a archivos profundo, S3 Intelligent-Tiering mueve automáticamente los objetos al nivel Acceso a archivos

profundo después de un mínimo de 180 días consecutivos sin acceso. Para ambos niveles, puede configurar la cantidad de días sin acceso según sus necesidades.

Las siguientes acciones constituyen un acceso que impide la jerarquización de sus objetos hasta el nivel Archive Access o el nivel Deep Archive Access:

- Descarga o copia de un objeto a través de la consola de Amazon S3.
- Invocación de CopyObject, UploadPartCopy o replicación de objetos con la replicación por lotes de S3. En estos casos, los objetos de origen de las operaciones de copia o replicación están agrupados en niveles.
- Invocación de GetObject, PutObject, RestoreObject, CompleteMultipartUpload, ListParts o SelectObjectContent.

Por ejemplo, si se accede a sus objetos a través de `SelectObjectContent` antes del número especificado de días sin acceso (por ejemplo, 180 días), esa acción reinicia el temporizador. Sus objetos no pasarán al nivel Archive Access ni al nivel Deep Archive Access hasta que el tiempo transcurrido desde la última solicitud `SelectObjectContent` alcance el número de días que haya especificado.

Si se obtiene acceso a un objeto del nivel Infrequent Access o Archive Instant Access, este se desplaza automáticamente al nivel Infrequent Access.

Las siguientes acciones constituyen un acceso que mueve automáticamente los objetos del nivel Infrequent Access o el nivel Archive Instant Access de vuelta al nivel Frequent Access:

- Descarga o copia de un objeto a través de la consola de Amazon S3.
- Invocación de CopyObject, UploadPartCopy o replicación de objetos con la replicación por lotes. En estos casos, los objetos de origen de las operaciones de copia o replicación están agrupados en niveles.
- Invocación de GetObject, PutObject, RestoreObject, CompleteMultipartUpload o ListParts.

Otras acciones no constituyen un acceso que mueve automáticamente los objetos del nivel Infrequent Access o el nivel Archive Instant Access de vuelta al nivel Frequent Access. La siguiente es una muestra, no una lista definitiva, de dichas acciones:

- Invocación de HeadObject, GetObjectTagging, PutObjectTagging, ListObjects, ListObjectsV2 o ListObjectVersions.

- La invocación de SelectObjectContent no constituye un acceso que eleve los objetos a un nivel Frequent Access. Además, no impide bajar los objetos del nivel Acceso frecuente al nivel Acceso poco frecuente y luego al nivel Acceso instantáneo a archivos.

Puede configurar S3 Intelligent-Tiering como clase de almacenamiento predeterminada para los datos recién creados especificando `INTELLIGENT-TIERING` en su encabezado de solicitud PutBucketIntelligentTieringConfiguration. S3 Intelligent-Tiering está diseñado para ofrecer una disponibilidad del 99.9 % y una durabilidad del 99.9999999 %.

> **Note**
>
> Si el tamaño del objeto es inferior a 128 KB, no se monitorea y no es elegible para los niveles automáticos. Los objetos más pequeños siempre se almacenan en la capa de acceso frecuente.

Capas de acceso de S3 Intelligent-Tiering

En la siguiente sección se explican los diferentes niveles de acceso automático y opcional. Cuando los objetos se mueven entre niveles de acceso, la clase de almacenamiento sigue siendo la misma (S3 Intelligent-Tiering).

Capa de acceso frecuente (automático)

Esta es la capa de acceso predeterminada en la que cualquier objeto creado o transitado a S3 Intelligent-Tiering comienza su ciclo de vida. Un objeto permanece en esta capa mientras se accede a él. El nivel Acceso frecuente proporciona baja latencia y alto rendimiento.

Capa de acceso poco frecuente (automático)

Si no se accede a un objeto durante 30 días consecutivos, el objeto se mueve a la capa de acceso poco frecuente. El nivel Acceso poco frecuente proporciona baja latencia y alto rendimiento.

Nivel Archive Instant Access (automático)

Si no se accede a un objeto durante 90 días consecutivos, el objeto se mueve al nivel Archive Instant Access. El nivel Acceso instantáneo a archivos proporciona baja latencia y alto rendimiento.

Capa de acceso al archivo (opcional)

S3 Intelligent-Tiering le proporciona la opción de activar el nivel Archive Access para los datos a los que se puede acceder de forma asíncrona. Después de la activación, el nivel Archive Access archiva automáticamente objetos a los que no se accedió durante un mínimo de 90 días consecutivos. Puede ampliar el tiempo de último acceso para archivar hasta un máximo de 730 días. El nivel Archive Access tiene el mismo rendimiento que la clase de almacenamiento S3 Glacier Flexible Retrieval.

Los tiempos de recuperación estándar para este nivel de acceso pueden oscilar entre 3 y 5 horas. Si inicia la solicitud de restauración mediante Operaciones por lotes de S3, la restauración se iniciará en cuestión de minutos. Para obtener más información sobre las opciones y los tiempos de recuperación, consulte the section called “Restauración de objetos desde los niveles S3 Intelligent-Tiering Archive Access o Deep Archive Access”.

> **Note**
>
> Active el nivel Archive Access solo durante 90 días si desea omitir el nivel Archive Instant Access. El nivel Acceso a archivos ofrece costos de almacenamiento ligeramente inferiores con tiempos de recuperación de minutos a horas. El nivel Archive Instant Access proporciona acceso en milisegundos y un alto rendimiento.

Capa de acceso de archivo profundo (opcional)

S3 Intelligent-Tiering le proporciona la opción de activar el nivel Deep Archive Access para datos a los que se puede acceder de forma asíncrona. Después de la activación, el nivel Deep Archive Access archiva automáticamente objetos a los que no se accedió durante un mínimo de 180 días consecutivos. Puede ampliar el tiempo de último acceso para archivar hasta un máximo de 730 días. La capa de acceso de archivo profundo tiene el mismo rendimiento que la clase de almacenamiento S3 Glacier Deep Archive.

La recuperación estándar de objetos de este nivel de acceso se produce en un plazo de 12 horas. Si inicia la solicitud de restauración mediante Operaciones por lotes de S3, la restauración se iniciará en un plazo de 9 horas. Para obtener más información sobre las opciones y los tiempos de recuperación, consulte the section called “Restauración de objetos desde los niveles S3 Intelligent-Tiering Archive Access o Deep Archive Access”.

> **Note**
>
> Active las capas de acceso a archivo y archivo profundo solo si la aplicación puede acceder de forma asíncrona a sus objetos. Si el objeto que está recuperando está almacenado en los niveles Archive Access o Deep Archive Access, primero deberá restaurar el objeto utilizando la operación `RestoreObject`.

Uso de S3 Intelligent-Tiering

Puede utilizar la clase de almacenamiento S3 Intelligent-Tiering para optimizar automáticamente los costes de almacenamiento. S3 Intelligent-Tiering ofrece un ahorro automático en los costes mediante la migración de datos en un nivel de objeto pormenorizado entre capas de acceso cuando los patrones de acceso cambian. Para los datos a los que se puede acceder de forma asíncrona, puede optar por habilitar el archivo automático dentro de la clase de almacenamiento de S3 Intelligent-Tiering utilizando AWS Management Console, AWS CLI o la API de Amazon S3.

Transición de datos a S3 Intelligent-Tiering

Existen dos formas de mover datos a S3 Intelligent-Tiering. Usted puede directamente PUT datos en S3 Intelligent-Tiering especificando `INTELLIGENT_TIERING` en el encabezado `x-amz-storage-class` o configurar el Ciclo de vida de S3 para cambiar de S3 Standard o S3 Standard-Infrequent Access a S3 Intelligent-Tiering.

Carga de datos a S3 Intelligent-Tiering con Direct Put

Cuando se carga un objeto a la clase de almacenamiento de S3 Intelligent-Tiering con la operación PUT de la API, se especifica S3 Intelligent-Tiering en el encabezado de la solicitud `x-amz-storage-class`.

La siguiente solicitud almacena la imagen,`my-image.jpg`, en el bucket `myBucket`. La solicitud utiliza el encabezado `x-amz-storage-class` para solicitar que el objeto se almacene con la clase de almacenamiento de 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
```

Transición de datos a S3 Intelligent-Tiering de S3 Standard o S3 Standard–Infrequent Access a través del ciclo de vida de S3

Puede agregar reglas a una configuración del ciclo de vida de S3 para indicar a Amazon S3 que pase objetos de una clase de almacenamiento a otro. Para obtener información sobre las transiciones admitidas y las restricciones relacionadas, consulte Transición de objetos con el ciclo de vida de S3.

Puede especificar configuraciones de Ciclo de vida de S3 en el nivel de bucket o de prefijo. En esta regla de configuración del ciclo de vida de S3, el filtro especifica un prefijo de clave (`documents/`). Por lo tanto, la regla se aplica a objetos con el prefijo de nombre de clave `documents/`, como `documents/doc1.txt` y `documents/doc2.txt`. La regla especifica una acción `Transition` que le indica a Amazon S3 que pase los objetos a la clase de almacenamiento de S3 Intelligent-Tiering 0 días después de su creación. En este caso, los objetos son elegibles para la transición a S3 Intelligent-Tiering a medianoche UTC después de su creación.

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>
```

Habilitación de los niveles S3 Intelligent-Tiering Archive Access o Deep Archive Access

Para obtener el costo de almacenamiento de datos más bajo, al que se puede acceder en minutos a horas, puede activar uno o ambos niveles de acceso a archivos creando una configuración de etiqueta de objeto, de prefijo o de bucket utilizando AWS Management Console, AWS CLI o la API de Amazon S3.

Uso de la consola de S3

Para habilitar el archivado automático de S3 Intelligent-Tiering

1. Inicie sesión en la AWS Management Console y abra la consola de Amazon S3 en https://console.aws.amazon.com/s3/.
2. En la lista Buckets (Buckets), elija el nombre del bucket en cuestión.
3. Seleccione Properties (Propiedades).
4. Vaya a la sección S3 Intelligent-Tiering Archive configurations (Configuraciones de archivo S3 Intelligent-Tiering) y elija Create configuration (Creación de configuración).
5. En la sección Archive configuration settings (Ajustes de configuración del archivo), especifique un nombre de configuración descriptivo para la configuración de archivo de S3 Intelligent-Tiering.
6. En Choose a configuration scope (Elegir un alcance de configuración), elija un alcance de configuración para usar. Opcionalmente, puede limitar el alcance de configuración a los objetos especificados dentro de un bucket mediante un prefijo compartido, una etiqueta de objeto o una combinación de ambos.
    a. Para limitar el alcance de la configuración, seleccione Limit the scope of this configuration using one or more filters (Limitar el alcance de esta configuración usando uno o más filtros).
    b. Para limitar el alcance de la configuración con un prefijo único, escriba el prefijo en Prefix (Prefijo).
    c. Para limitar el alcance de la configuración mediante etiquetas de objeto, seleccione Add tad (Agregar etiqueta) e introduzca un valor para Clave.
7. En Status (Estado), seleccione Enable (Habilitar).
8. En la sección Archive settings (Configuración de archivo), seleccione una o cambas de las capas de acceso de archivo para habilitar.
9. Seleccione Create (Crear).

Utilización de la AWS CLI

Puede utilizar los siguientes comandos de la AWS CLI para administrar configuraciones de S3 Intelligent-Tiering.

- `delete-bucket-intelligent-tiering-configuration`
- `get-bucket-intelligent-tiering-configuration`
- `list-bucket-intelligent-tiering-configurations`
- `put-bucket-intelligent-tiering-configuration`

Para obtener instrucciones acerca de cómo configurar la AWS CLI, consulte Desarrollo con Amazon S3 mediante la AWS CLI.

Cuando se usa la AWS CLI, no puede especificar la configuración como archivo XML. Debe especificar el JSON en su lugar. El siguiente es un ejemplo de configuración de XML S3 Intelligent-Tiering de XML y JSON equivalente que puede especificar en un comando de la AWS CLI.

En el siguiente ejemplo se establece una configuración de S3 Intelligent-Tiering en el bucket especificado.

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>
```

Uso de la operación PUT de la API

Puede utilizar la operación `PutBucketIntelligentTieringConfiguration` para un bucket especificado y hasta 1000 configuraciones de S3 Intelligent-Tiering por cada bucket. Puede definir qué objetos dentro de un bucket son aptos para las capas de acceso a archivos utilizando un prefijo compartido o una etiqueta de objeto. Los prefijos compartidos o las etiquetas de objetos permiten alinear las aplicaciones de negocios específicas, los flujos de trabajo o las organizaciones internas. También tiene la flexibilidad de habilitar la capa de acceso a archivo, la capa de acceso a archivo profundo, o ambas.

Introducción a S3 Intelligent-Tiering

Para obtener más información sobre cómo usar S3 Intelligent-Tiering, consulte el Tutorial: Introducción al uso de S3 Intelligent-Tiering.

Administración de S3 Intelligent-Tiering

La clase de almacenamiento S3 Intelligent-Tiering ofrece ahorro automático en los costos de almacenamiento en tres niveles de acceso de baja latencia y alto rendimiento. También ofrece capacidades de archivo opcionales para ayudarle a obtener los costos de almacenamiento más bajos en la nube para datos a los que se puede acceder en minutos a horas. La clase de almacenamiento S3 Intelligent-Tiering admite todas las características de Amazon S3, incluidas las siguientes:

- S3 Inventory, para verificar el nivel de acceso de los objetos
- Replicación de S3, para replicar datos en cualquier Región de AWS
- S3 Storage Lens, para consultar las métricas de actividad y uso del almacenamiento
- Cifrado del servidor, para proteger datos de objetos
- Bloqueo de objetos de S3, para evitar la eliminación accidental de datos
- AWS PrivateLink, para acceder a Amazon S3 a través de un punto de conexión privado en una nube privada virtual (VPC)

Identificación en qué objetos de capa de acceso de S3 Intelligent-Tiering se almacenan los objetos

Para obtener una lista de sus objetos y sus metadatos correspondientes, incluido su nivel de acceso de S3 Intelligent-Tiering, puede utilizar the section called “Administración del inventario”.

Inventario de S3 proporciona archivos de salida CSV, ORC o Parquet que enumeran sus objetos y sus metadatos correspondientes. Puede recibir estos informes de inventario de forma diaria o semanal para un bucket de Amazon S3 o un prefijo compartido. (El prefijo compartido hace referencia a objetos que tienen nombres que comienzan con la misma cadena).

Visualización del estado de archivo de un objeto dentro de S3 Intelligent-Tiering

Para recibir un aviso cuando un objeto en la clase de almacenamiento de S3 Intelligent-Tiering se ha movido al nivel Acceso a archivos o al nivel Acceso a archivos profundo, puede configurar Notificaciones de eventos de S3. Para obtener más información, consulte Habilitación de notificaciones de eventos.

Amazon S3 puede publicar notificaciones de eventos en un tema de Amazon Simple Notification Service (Amazon SNS), una cola de Amazon Simple Queue Service (Amazon SQS) o una función de AWS Lambda. Para obtener más información, consulte Notificaciones de eventos de Amazon S3.

El siguiente mensaje es un ejemplo de un mensaje que Amazon S3 envía para publicar un evento de `s3:IntelligentTiering`. Para obtener más información, consulte the section called “Estructura de mensaje de evento”.

```
{
   "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"
            }
      }
   ]
}
```

También puede utilizar una solicitud de objeto HEAD para ver el estado del archivo de un objeto. Si un objeto se almacena mediante la clase de almacenamiento de S3 Intelligent-Tiering y se encuentra en uno de los niveles de archivo, la respuesta del objeto HEAD mostrará el nivel de archivo actual. Para mostrar el nivel de archivo, la solicitud utiliza el encabezado x-amz-archive-status.

La siguiente solicitud de objeto HEAD devuelve los metadatos de un objeto (en este caso, *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=
```

También puede utilizar solicitudes de objeto HEAD para monitorear el estado de una solicitud `restore-object`. Si la restauración del archivo está en curso, la respuesta del objeto HEAD incluirá el encabezado x-amz-restore.

A continuación, se muestra un ejemplo de respuesta de objeto HEAD que muestra un objeto archivado mediante S3 Intelligent-Tiering con una solicitud de restauración en curso.

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
```

Restauración de objetos desde los niveles S3 Intelligent-Tiering Archive Access o Deep Archive Access

Para acceder a los objetos de los niveles S3 Intelligent-Tiering Archive Access y Deep Archive Access, debe iniciar una solicitud de restauración y, a continuación, esperar a que el objeto pase al nivel Acceso frecuente. Para obtener más información acerca de objetos archivados, consulte the section called “Trabajar con objetos archivados”.

Cuando se restaura desde las capas de acceso de archivo de o acceso profundo, el objeto vuelve a pasar a la capa de acceso frecuente. Después, si no se accede al objeto durante 30 días consecutivos, se mueve automáticamente al nivel Infrequent Access. Después, tras un mínimo de 90 días consecutivos sin acceso, el objeto pasa al nivel Acceso a archivos. Tras un mínimo de 180 días consecutivos sin acceso, el objeto pasa al nivel Acceso a archivos profundo. Para obtener más información, consulte the section called “Cómo funciona S3 Intelligent-Tiering”.

Puede restaurar un objeto archivado mediante la consola de Amazon S3, Operaciones por lotes de S2, la API de REST de Amazon S3, los SDK de AWS o la AWS Command Line Interface (AWS CLI). Para obtener más información, consulte the section called “Trabajar con objetos archivados”.

Administración del ciclo de vida del almacenamiento

Para administrar los objetos de manera que se almacenen de forma económica durante todo su ciclo de vida, cree una configuración de Amazon S3 Lifecycle. La configuración de Amazon S3 Lifecycle

es un conjunto de reglas que definen acciones que Amazon S3 aplica a un grupo de objetos. Existen dos tipos de acciones:

- Acciones de transición: estas acciones definen el momento en que los objetos pasan a otra clase de almacenamiento. Por ejemplo, podría decidir pasar objetos a la clase de almacenamiento S3 Standard-IA 30 días después de su creación o archivar objetos en la clase de almacenamiento S3 Glacier Flexible Retrieval un año después de su creación. Para obtener más información, consulte Uso de las clases de almacenamiento de Amazon S3.

  Hay costos asociados con las solicitudes de transición de ciclo de vida. Para obtener información acerca de los precios, consulte Precios de Amazon S3.
- Acciones de vencimiento: estas acciones definen el momento en que vencen los objetos. Amazon S3 elimina automáticamente los objetos que han vencido.

  Los costos de vencimiento del ciclo de vida dependen de cuándo elige provocar el vencimiento de los objetos. Para obtener más información, consulte Vencimiento de objetos.

> **Important**
>
> No puede usar una política de bucket para evitar eliminaciones ni transiciones mediante una regla de S3 Lifecycle. Por ejemplo, aunque la política de bucket deniegue todas las acciones a todas las entidades principales, la configuración de S3 Lifecycle seguirá funcionando con normalidad.

Objetos existentes y nuevos

Cuando añade una configuración de ciclo de vida a un bucket, las reglas de configuración se aplican a los objetos existentes y a los objetos que añade posteriormente. Por ejemplo, si hoy agrega una regla de configuración de Lifecycle con una acción de vencimiento que causa que objetos expiren 30 días después de la creación, Amazon S3 pondrá en cola de eliminación cualquier objeto existente con más de 30 días de antigüedad.

Cambios en la facturación

Si hay algún retraso entre el momento en que un objeto pasa a ser apto para una acción del ciclo de vida y cuando Amazon S3 transfiere o caduca el objeto, los cambios de facturación se aplican tan pronto como el objeto es apto para la acción del ciclo de vida. Por ejemplo, si un objeto

está programado para vencer y Amazon S3 no lo finaliza inmediatamente, no se le cobrará el almacenamiento después del tiempo de vencimiento.

La única excepción a este comportamiento es si tiene una regla de ciclo de vida para realizar la transición a la clase de almacenamiento S3 Intelligent-Tiering. En ese caso, los cambios en la facturación no se producen hasta que el objeto haya pasado a S3 Intelligent-Tiering.

Para obtener más información acerca de las reglas de S3 Lifecycle, consulte Elementos de configuración del ciclo de vida.

Supervisión del efecto de las reglas del ciclo de vida

Para supervisar el efecto de las actualizaciones realizadas por las reglas del ciclo de vida activo, consulte the section called “¿Cómo puedo supervisar las acciones que se llevan a cabo según mis reglas de ciclo de vida?”.

Administración del ciclo de vida de los objetos

Defina reglas de configuración de S3 Lifecycle para los objetos que tienen un ciclo de vida bien definido. Por ejemplo:

- Si carga logs periódicos en un bucket, es posible que la aplicación los necesite durante una semana o un mes. Una vez transcurrido ese tiempo, es posible que desee eliminarlos.
- Se obtiene acceso a algunos documentos con frecuencia durante un periodo limitado. Posteriormente, se obtendrá acceso a ellos con poca frecuencia. En algún momento, es posible que no necesite acceso en tiempo real a estos objetos, pero la organización o las normativas pueden requerir su archivado durante un periodo específico. Transcurrido dicho periodo, podrá eliminarlos.
- Es posible que desee cargar algunos tipos de datos a Amazon S3 para su archivado. Por ejemplo, podría archivar medios digitales, registros financieros y sanitarios, datos de secuencias genómicas sin procesar, backups de bases de datos a largo plazo y datos que deben conservarse por motivos de conformidad normativa.

Con las reglas de configuración de S3 Lifecycle, puede indicarle a Amazon S3 que pase los objetos a otras clases de almacenamiento más económicas, que los archive o que los elimine.

Creación de una configuración del ciclo de vida

Una configuración de S3 Lifecycle, que se guarda en un archivo XML, consta de un conjunto de reglas con acciones predefinidas que desea que Amazon S3 realice en los objetos durante su vida útil.

También puede crear una configuración de ciclo de vida mediante la consola de Amazon S3, la API de REST, los AWS SDK y la AWS Command Line Interface (AWS CLI). Para obtener más información, consulte Configuración de un ciclo de vida en un bucket.

Amazon S3 proporciona un conjunto de operaciones de la API REST para administrar la configuración del ciclo de vida de un bucket. Amazon S3 almacena la configuración como un subrrecurso del ciclo de vida que se asocia a su bucket. Para obtener más información, consulte los siguientes temas:

- PutBucketLifecycleConfiguration
- GetBucketLifecycleConfiguration
- DeleteBucketLifecycle

Para obtener más información acerca de cómo crear una configuración del ciclo de vida, consulte los temas siguientes:

Temas

- Transición de objetos con Amazon S3 Lifecycle
- Vencimiento de objetos
- Configuración de un ciclo de vida en un bucket
- Configuraciones del ciclo de vida y otras configuraciones del bucket
- Configurar notificaciones de eventos de Lifecycle
- Elementos de configuración del ciclo de vida
- Ejemplos de configuración de S3 Lifecycle

Transición de objetos con Amazon S3 Lifecycle

Puede agregar reglas a una configuración de S3 Lifecycle para indicar a Amazon S3 que pase objetos a otra clase de almacenamiento de Amazon S3. Para obtener más información acerca de las clases de almacenamiento, consulte Uso de las clases de almacenamiento de Amazon S3. A

continuación se muestran algunos ejemplos de cuándo podría utilizar las configuraciones del ciclo de vida de S3 de esta forma:

- Si sabe que se obtiene acceso a determinados objetos con poca frecuencia, puede pasarlos a la clase de almacenamiento S3 Standard-IA.
- Se recomienda archivar los objetos a los que no necesita obtener acceso en tiempo real en las clases de almacenamiento S3 Glacier Flexible Retrieval o S3 Glacier Deep Archive.

Objetos existentes y nuevos

Cuando añade una configuración de ciclo de vida a un bucket, las reglas de configuración se aplican a los objetos existentes y a los objetos que añade posteriormente. Por ejemplo, si hoy agrega una regla de configuración de Lifecycle con una acción de transición que causa que objetos con un prefijo específico se transfieran a una clase de almacenamiento diferente 30 días después de la creación, Amazon S3 pondrá en cola de transición cualquier objeto existente con más de 30 días de antigüedad y que tenga el prefijo especificado.

> **Important**
>
> No puede usar una política de bucket para evitar eliminaciones ni transiciones mediante una regla de S3 Lifecycle. Por ejemplo, aunque la política de bucket deniegue todas las acciones a todas las entidades principales, la configuración de S3 Lifecycle seguirá funcionando con normalidad.

Metadatos de objetos

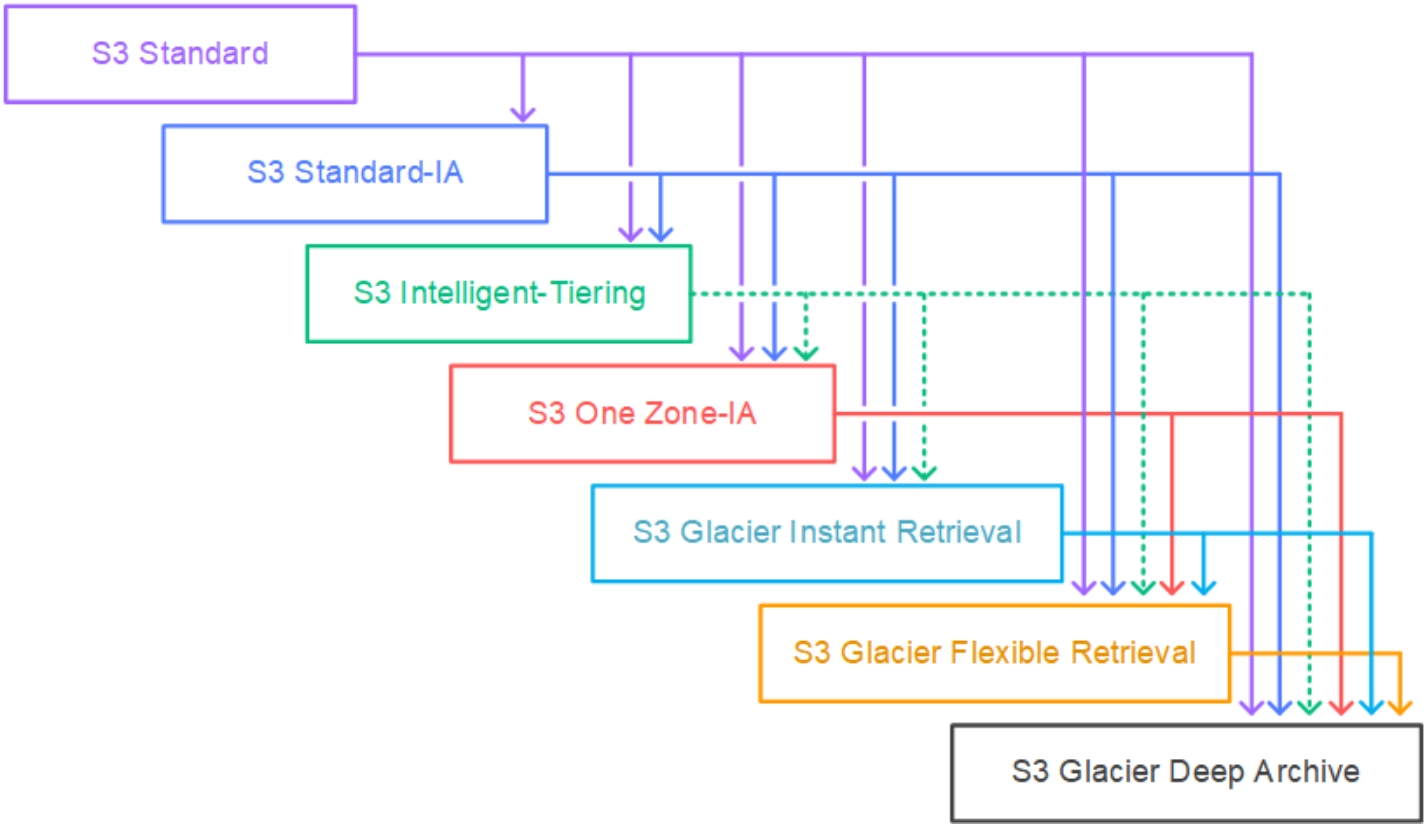
Para las acciones de Lifecycle, se comprueban dos tipos de metadatos de objetos: Bloqueo de objetos de S3 y un estado de replicación de `Pending`. Si Bloqueo de objetos de S3 está configurado en un objeto, Lifecycle no hace que venzan las versiones no actuales de ese objeto. Esta validación solo se comprueba cuando vence una clave con control de versiones o una versión no actual. Además, las claves con estados de replicación pendientes no están en transición ni vencen. Esta validación solo se aplica a las claves con control de versiones (tanto las versiones actuales como las no actuales).

Transiciones admitidas y limitaciones relacionadas

En una configuración del ciclo de vida de S3, puede definir reglas para pasar objetos de una clase de almacenamiento a otra para ahorrar costos de almacenamiento. Cuando desconoce los patrones

de acceso de los objetos o si los patrones de acceso cambian con el tiempo, es posible realizar la transición de los objetos a la clase de almacenamiento S3 Intelligent-Tiering para un ahorro automático de los costos. Para obtener más información acerca de las clases de almacenamiento, consulte Uso de las clases de almacenamiento de Amazon S3.

Amazon S3 admite un modelo en cascada para las transiciones entre clases de almacenamiento, tal y como se muestra en el siguiente diagrama.



Transiciones del ciclo de vida admitidas

Amazon S3 admite las siguientes transiciones de ciclo de vida entre clases de almacenamiento con una configuración de S3 Lifecycle.

Puede realizar la transición de lo siguiente:

- La clase de almacenamiento S3 Standard a cualquier otra clase de almacenamiento.
- La clase de almacenamiento S3 Standard-IA a las clases de almacenamiento S3 Intelligent-Tiering, S3 One Zone-IA, S3 Glacier Instant Retrieval, S3 Glacier Flexible Retrieval o S3 Glacier Deep Archive.
- La clase de almacenamiento S3 Intelligent-Tiering a las clases de almacenamiento S3 One Zone-IA, S3 Glacier Instant Retrieval, S3 Glacier Flexible Retrieval o S3 Glacier Deep Archive.

> **Note**
>
> Hay algunas excepciones en la transición de objetos de la clase de almacenamiento S3 Intelligent-Tiering a S3 One Zone-IA y algunas clases de almacenamiento S3 Glacier. Para obtener más información, consulte the section called “Transiciones del ciclo de vida no admitidas”.

- Cualquier clase de almacenamiento S3 One Zone-IA a las clases de almacenamiento S3 Glacier Flexible Retrieval o S3 Glacier Deep Archive.
- La clase de almacenamiento S3 Glacier Instant Retrieval a las clases de almacenamiento S3 Glacier Flexible Retrieval o S3 Glacier Deep Archive.
- La clase de almacenamiento S3 Glacier Flexible Retrieval a la clase de almacenamiento S3 Glacier Deep Archive.
- Cualquier clase de almacenamiento a las clases de almacenamiento S3 Glacier Deep Archive.

> **Note**
>
> No hay cargos por recuperación de datos para las transiciones del ciclo de vida. Sin embargo, existen cargos por ingesta por solicitud cuando se utilizan reglas `PUT`, `COPY` o de ciclo de vida para mover datos a cualquier clase de almacenamiento de S3. Plantéese el costo de ingesta o de transición antes de mover objetos a cualquier clase de almacenamiento. Para obtener más información acerca de las consideraciones sobre costes, consulte Precios de Amazon S3.

Transiciones del ciclo de vida no admitidas

Amazon S3 no admite ninguna de las siguientes transiciones de ciclo de vida.

No se puede realizar la transición de lo siguiente:

- Cualquier clase de almacenamiento a la clase de almacenamiento S3 Standard.
- Cualquier clase de almacenamiento a la clase de almacenamiento de redundancia reducida (RRS).
- La clase de almacenamiento S3 One Zone-IA a las clases de almacenamiento S3 Intelligent-Tiering, S3 Standard-IA o S3 Glacier Instant Retrieval.

- La clase de almacenamiento S3 Intelligent-Tiering (todos los niveles) a la clase de almacenamiento S3 Standard-IA.
- La clase de almacenamiento S3 Intelligent-Tiering, nivel Archive Instant Access, a S3 One Zone-IA.
- La clase de almacenamiento S3 Intelligent-Tiering, nivel Archive Access, a S3 One Zone-IA o S3 Glacier Instant Retrieval.
- La clase de almacenamiento S3 Intelligent-Tiering, nivel Deep Archive Access, a S3 One Zone-IA, S3 Glacier Instant Retrieval o S3 Glacier Flexible Retrieval.

Restricciones

Las transiciones de clases de almacenamiento de ciclo de vida tienen las siguientes limitaciones:

Tamaño de objeto y transiciones de S3 Standard o S3 Standard-IA a S3 Intelligent-Tiering, S3 Standard-IA o S3 One Zone-IA.

Cuando se pasan objetos de las clases de almacenamiento S3 Standard o S3 Standard-IA a S3 Intelligent-Tiering (Capas inteligentes de S3), S3 Standard-IA o S3 One Zone-IA, se aplican las siguientes restricciones de tamaño de objeto:

- Objetos grandes: para las siguientes transiciones, la transición de objetos grandes supone un beneficio económico:
  - De las clases de almacenamiento S3 Standard o S3 Standard-IA a S3 Intelligent-Tiering (Capas inteligentes de S3).
  - De las clases de almacenamiento S3 Standard a S3 Standard-IA o S3 One Zone-IA.
- Objetos de menos de 128 KiB: para las siguientes transiciones, Amazon S3 no realiza la transición de objetos que tienen menos de 128 KiB:
  - De las clases de almacenamiento S3 Standard o S3 Standard-IA a S3 Intelligent-Tiering o S3 Glacier Instant Retrieval.
  - De las clases de almacenamiento S3 Standard a S3 Standard-IA o S3 One Zone-IA.

> **Note**
>
> Puede filtrar las reglas del ciclo de vida según el tamaño del objeto.

Important

Cuando tiene varias reglas en una configuración de S3 Lifecycle, un objeto puede reunir los requisitos para varias acciones de S3 Lifecycle realizadas el mismo día. En tales casos, Amazon S3 sigue estas reglas generales:

- La eliminación permanente prevalece sobre la transición.
- La transición prevalece sobre la creación de marcadores de eliminación.
- Cuando un objeto es elegible para una transición S3 Glacier Flexible Retrieval y S3 Standard-IA (o S3 One Zone-IA), Amazon S3 elige la transición S3 Glacier Flexible Retrieval.

Para ver ejemplos, consulte Ejemplo 5: superposición de filtros, conflictos entre acciones de ciclo de vida y lo que hace Amazon S3 con buckets sin control de versiones.

Días mínimos para la transición a S3 Standard-IA o S3 One Zone-IA

Antes de pasar objetos a S3 Standard-IA o S3 One Zone-IA, debe almacenarlos durante al menos 30 días en Amazon S3. Por ejemplo, no puede crear una regla de ciclo de vida para pasar objetos a la clase de almacenamiento S3 Standard-IA un día después de su creación. Amazon S3 no admite esta transición durante los primeros 30 días porque se suele obtener acceso a los objetos más nuevos con mayor frecuencia o estos se eliminan antes de lo que corresponde para las clases de almacenamiento S3 Standard-IA o S3 One Zone-IA.

Asimismo, si pasa objetos no actuales (en los buckets con control de versiones), solo podrá pasarlos a las clases de almacenamiento S3 Standard-IA o S3 One Zone-IA si cumplen la condición de ser no actuales durante, al menos, 30 días. Para obtener una lista de la duración mínima de almacenamiento de todas las clases de almacenamiento, consulte Comparación de las clases de almacenamiento de Amazon S3.

Cargo mínimo de almacenamiento de 30 días para S3 Standard-IA y S3 One Zone-IA

Las clases de almacenamiento S3 Standard-IA y S3 One Zone-IA tienen un cargo mínimo de almacenamiento de 30 días. Por lo tanto, no puede especificar una única regla de ciclo de vida tanto para una transición S3 Standard-IA o S3 One Zone-IA como para una transición S3 Glacier Flexible Retrieval o S3 Glacier Deep Archive si la transición S3 Glacier Flexible Retrieval o S3 Glacier Deep Archive se produce menos de 30 días después de la transición S3 Standard-IA o S3 One Zone-IA.

Este mismo cargo mínimo de 30 días se aplica cuando se especifica una transición del almacenamiento de S3 Standard-IA a S3 One Zone-IA. Puede especificar dos reglas para lograrlo, pero deberá pagar los cargos de almacenamiento mínimos. Para obtener más información acerca de las consideraciones sobre costes, consulte Precios de Amazon S3.

Administrar el ciclo de vida completo de un objeto

Puede combinar estas acciones de S3 Lifecycle para administrar el ciclo de vida completo de un objeto. Por ejemplo, supongamos que los objetos que crea tienen un ciclo de vida bien definido. Al principio, se obtiene acceso a los objetos con frecuencia durante un periodo de 30 días. Posteriormente, se obtiene acceso a los objetos con poca frecuencia durante un periodo máximo de 90 días. Transcurrido ese tiempo, los objetos ya no son necesarios, por lo que podría archivarlo o eliminarlos.

En esta situación, puede crear una regla de S3 Lifecycle en la que especifique la acción de transición inicial a la clase de almacenamiento S3 Intelligent-Tiering, S3 Standard-IA o S3 One Zone-IA, otra acción de transición al almacenamiento S3 Glacier Flexible Retrieval para el archivo y una acción de vencimiento. A medida que se mueven los objetos de una clase de almacenamiento a otra, se ahorra en costos de almacenamiento. Para obtener más información acerca de las consideraciones sobre costes, consulte Precios de Amazon S3.

Transición a las clases de almacenamiento S3 Glacier Flexible Retrieval y S3 Glacier Deep Archive (archivo de objetos)

Con una configuración de S3 Lifecycle, puede realizar una transición de objetos a las clases de almacenamiento S3 Glacier Flexible Retrieval o S3 Glacier Deep Archive para archivarlos. Si elige la clase de almacenamiento S3 Glacier Flexible Retrieval o S3 Glacier Deep Archive, sus objetos permanecen en Amazon S3. No puede acceder a ellos directamente a través del servicio independiente de Amazon S3 Glacier. Para obtener más información general sobre S3 Glacier, consulte Qué es Amazon S3 Glacier en la Guía para desarrolladores de Amazon S3 Glacier.

Antes de archivar objetos, lea las siguientes secciones, donde encontrará consideraciones pertinentes.

Consideraciones generales

A continuación, se proporcionan consideraciones generales para que tenga en cuenta antes de archivar objetos:

- Los objetos cifrados siguen estando cifrados durante todo el proceso de transición de la clase de almacenamiento.
- Los objetos que se almacenan en las clases S3 Glacier Flexible Retrieval o S3 Glacier Deep Archive no están disponibles en tiempo real.

  Los objetos archivados son objetos de Amazon S3, pero antes de poder obtener acceso a un objeto archivado, debe primero restaurar una copia temporal de este. La copia del objeto restaurado solo está disponible durante el tiempo que especifique en la solicitud de restauración. Después de ese periodo, Amazon S3 elimina la copia temporal y el objeto permanece archivado en S3 Glacier Flexible Retrieval.

  Puede restaurar un objeto usando la consola de Amazon S3 o mediante programación con las bibliotecas de encapsulamiento de AWS SDK o la API REST de Amazon S3 en su código. Para obtener más información, consulte Restauración de un objeto archivado.
- Los objetos almacenados en la clase de almacenamiento S3 Glacier Flexible Retrieval solo se pueden pasar a la clase de almacenamiento S3 Glacier Deep Archive.

  Puede utilizar una regla de configuración de S3 Lifecycle para convertir la clase de almacenamiento de un objeto de S3 Glacier Flexible Retrieval a S3 Glacier Deep Archive solamente. Si desea cambiar la clase de almacenamiento de un objeto almacenado en S3 Glacier Flexible Retrieval a otra clase de almacenamiento que no sea S3 Glacier Deep Archive, debe usar primero la operación de restauración para hacer una copia temporal del objeto. A continuación, utilice la operación de copia para sobrescribir el objeto especificando S3 Standard, S3 Intelligent-Tiering (Capas inteligentes de S3), S3 Standard-IA, S3 One Zone-IA o Reduced Redundancy (Redundancia reducida) como clase de almacenamiento.
- La transición de objetos a la clase de almacenamiento S3 Glacier Deep Archive es unidireccional.

  No puede usar una regla de configuración de S3 Lifecycle para convertir la clase de almacenamiento de un objeto de S3 Glacier Deep Archive a cualquier otra clase de almacenamiento. Si desea cambiar la clase de almacenamiento de un objeto archivado a otra clase de almacenamiento, debe usar primero la operación de restauración para hacer una copia temporal del objeto. A continuación, utilice la operación de copia para sobrescribir el objeto especificando S3 Standard, S3 Intelligent-Tiering, S3 Standard-IA, S3 One Zone-IA, S3 Glacier Instant Retrieval, S3 Glacier Flexible Retrieval o Reduced Redundancy (redundancia reducida) como clase de almacenamiento.

> **Note**
>
> La operación de copia de objetos restaurados no se admite en la consola de Amazon S3 para los objetos de las clases de almacenamiento S3 Glacier Flexible Retrieval o S3 Glacier Deep Archive. Para este tipo de operación de copia, utilice la AWS Command Line Interface (AWS CLI), los SDK de AWS o la API de REST.

Los objetos que se almacenan en las clases de almacenamiento S3 Glacier Flexible Retrieval y S3 Glacier Deep Archive solo son visibles y están disponibles mediante Amazon S3. No están disponibles a través del servicio Amazon S3 Glacier independiente.

Estos son objetos de Amazon S3 y solo puede obtener acceso a ellos por medio de la consola de Amazon S3 o la API de Amazon S3. No puede obtener acceso a los objetos archivados a través de la consola de Amazon S3 Glacier independiente ni de la API de Amazon S3 Glacier.

Consideraciones sobre costos

Si tiene previsto archivar datos a los que accede con poca frecuencia durante un periodo de meses o años, las clases de almacenamiento S3 Glacier Flexible Retrieval y S3 Glacier Deep Archive pueden reducir los costos de almacenamiento. Sin embargo, para asegurarse de que la clase de almacenamiento S3 Glacier Flexible Retrieval o S3 Glacier Deep Archive es apropiada para usted, debe considerar lo siguiente:

- Cargos generales de almacenamiento: cuando realiza la transición de objetos a la clase de almacenamiento S3 Glacier Flexible Retrieval o S3 Glacier Deep Archive, se agrega una cantidad fija de almacenamiento a cada objeto para tener capacidad para los metadatos de manera que se pueda administrar el objeto.
  - Por cada objeto que se archiva en S3 Glacier Flexible Retrieval o S3 Glacier Deep Archive, Amazon S3 utiliza 8 KB de almacenamiento para el nombre del objeto y otros metadatos. Amazon S3 almacena estos metadatos para que pueda obtener una lista en tiempo real de los objetos archivados por medio de la API de Amazon S3. Para obtener más información, consulte Get Bucket (List Objects). Por este almacenamiento adicional se aplican las tarifas de S3 Standard.
  - Por cada objeto que se archiva en S3 Glacier Flexible Retrieval o S3 Glacier Deep Archive, Amazon S3 agrega 32 KB de almacenamiento para el índice y los metadatos relacionados.

Estos datos adicionales son necesarios para identificar y restaurar su objeto. Por este almacenamiento adicional se aplican las tarifas de S3 Glacier Flexible Retrieval o S3 Glacier Deep Archive.

Si archiva objetos pequeños, tenga en cuenta estos cargos de almacenamiento. Asimismo, considere la posibilidad de agregar muchos objetos pequeños a una cantidad más pequeña de objetos grandes para reducir los costos generales.

- Cantidad de días prevista para tener los objetos archivados: S3 Glacier Flexible Retrieval y S3 Glacier Deep Archive son soluciones de archivo a largo plazo. El periodo mínimo de almacenamiento es de 90 días para la clase de almacenamiento S3 Glacier Flexible Retrieval y de 180 para S3 Glacier Deep Archive. La eliminación de datos que están archivados en Amazon S3 Glacier no genera cargos si los objetos que elimina se archivan durante más tiempo que el periodo mínimo de almacenamiento. Si elimina o sobrescribe un objeto archivado antes de que transcurra el periodo mínimo, Amazon S3 aplica una tarifa de eliminación anticipada prorrateada. Para obtener información sobre la tarifa de eliminación anticipada, consulte la pregunta “¿Cómo se me cobrará por eliminar objetos de Amazon S3 Glacier que tienen menos de 90 días?” en Preguntas frecuentes sobre Amazon S3.
- Cargos de solicitud de transición a S3 Glacier Flexible Retrieval y S3 Glacier Deep Archive: cada objeto que pasa a la clase de almacenamiento S3 Glacier Flexible Retrieval o S3 Glacier Deep Archive constituye una solicitud de transición. Se aplica un costo para cada solicitud. Si tiene previsto pasar una cantidad grande de objetos, tenga en cuenta los costos de solicitud. Si va a archivar una combinación de objetos que incluye objetos pequeños, especialmente aquellos de menos de 128 KB, le recomendamos que utilice el filtro de tamaño de los objetos del ciclo de vida para filtrar los objetos pequeños de la transición y reducir los costos de las solicitudes. S3 Glacier Flexible Retrieval y S3 Glacier Deep Archive no bloquean automáticamente la transición de objetos de menos de 128 KB.
- Cargos de restauración de datos de S3 Glacier Flexible Retrieval y S3 Glacier Deep Archive: S3 Glacier Flexible Retrieval y S3 Glacier Deep Archive están diseñadas para archivar a largo plazo datos a los que se obtiene acceso con poca frecuencia. Para obtener información acerca de los cargos de restauración de datos, consulte la pregunta “¿Cuánto cuesta recuperar datos de Amazon S3 Glacier?” en Preguntas frecuentes sobre Amazon S3. Para obtener información acerca de cómo restaurar datos de Amazon S3 Glacier, consulte Restauración de un objeto archivado.

Cuando archiva objetos en Amazon S3 Glacier por medio de la administración de S3 Lifecycle, Amazon S3 realiza la transición de estos objetos de manera asincrónica. Es posible que haya un retraso entre la fecha de transición de la regla de configuración de S3 Lifecycle y la fecha de

la transición física. Se le aplican los precios de Amazon S3 Glacier según la fecha de transición especificada en la regla. Para obtener más información, consulte la sección Amazon S3 Glacier de las preguntas frecuentes de Amazon S3.

En la página de detalles del producto Amazon S3 se proporciona información sobre precios y ejemplos de cálculos para el archivo de objetos de Amazon S3. Para obtener más información, consulte los siguientes temas:

- "¿Cómo se calcula el costo de almacenamiento para los objetos de Amazon S3 archivados en Amazon S3 Glacier?" en Preguntas frecuentes sobre Amazon S3.
- "¿Cómo se cobra la eliminación de objetos de Amazon S3 Glacier que tengan menos de 90 días?" en Preguntas frecuentes sobre Amazon S3.
- "¿Cuánto cuesta recuperar datos de Amazon S3 Glacier?" en Preguntas frecuentes sobre Amazon S3.
- Precios de Amazon S3 para conocer los costes de almacenamiento de las diferentes clases de almacenamiento.

Restaurar objetos archivados

No se puede obtener acceso a los objetos archivados en tiempo real. Debe primero iniciar una solicitud de restauración y luego esperar hasta que haya una copia temporal del objeto disponible durante el periodo que especifique en la solicitud. Después de recibir una copia temporal del objeto restaurado, la clase de almacenamiento del objeto sigue siendo S3 Glacier Flexible Retrieval o S3 Glacier Deep Archive. (Una solicitud de operación de la API de HeadObject o GetObject devolverá S3 Glacier Flexible Retrieval o S3 Glacier Deep Archive como clase de almacenamiento).

> **Note**
>
> Cuando restaura un archivo, paga el archivo (la tarifa de S3 Glacier Flexible Retrieval o S3 Glacier Deep Archive) y una copia que restauró de manera temporal (tarifa de almacenamiento de S3 Standard). Para obtener información acerca de los precios, consulte Precios de Amazon S3.

Puede restaurar una copia del objeto mediante la consola de Amazon S3 o programación. Amazon S3 solo procesa una solicitud de restauración a la vez por cada objeto. Para obtener más información, consulte Restauración de un objeto archivado.

Vencimiento de objetos

Cuando un objeto llega al final de su vida útil según su configuración de ciclo de vida, Amazon S3 realizará una acción según el estado de control de versiones de S3 en el que se encuentre el bucket.

- Bucket no versionado: Amazon S3 coloca el objeto en cola para eliminarlo y lo hace de manera asincrónica, por lo que se elimina de forma permanente.
- Bucket con el control de versiones activado: si la versión actual del objeto no es un marcador de eliminación, Amazon S3 añade un marcador de eliminación con un ID de versión exclusivo. Esto convierte la versión actual en no actual, y el marcador de eliminación en la versión actual.
- Bucket con el control de versiones suspendido: Amazon S3 crea un marcador de eliminación con un ID de versión nulo. Este marcador de eliminación sustituye cualquier versión del objeto con un ID de versión null en la jerarquía de versiones, lo que elimina el objeto de manera eficaz.

Para un bucket con control de versiones (un bucket con el control de versiones habilitado o suspendido), hay varias consideraciones que se deben tener en cuenta en relación con cómo administra Amazon S3 la acción de expiración. En el caso de los buckets con el control de versiones activado o suspendido, se aplica lo siguiente:

- La caducidad del objeto solo se aplica a la versión actual del objeto (no tiene ningún impacto sobre las versiones del objeto no actuales).
- Amazon S3 no realiza ninguna acción si hay una o varias versiones del objeto y el marcador de eliminación es la versión actual.
- Si la versión actual del objeto es la única versión del objeto y, además, es un marcador de eliminación (también denominado marcador de eliminación de objeto vencido, en el que todas las versiones del objeto se han eliminado y solo queda un marcador de eliminación), Amazon S3 elimina el marcador de eliminación del objeto vencido. También puede usar la acción de vencimiento para ordenar a Amazon S3 que elimine los marcadores de eliminación de objeto vencidos. Por ejemplo, consulte Ejemplo 7: eliminar marcadores de eliminación de objetos que vencieron.
- Puede usar el elemento de acción `NoncurrentVersionExpiration` para indicar a Amazon S3 que elimine de forma permanente versiones no actuales de objetos. Estos objetos eliminados no se pueden recuperar. Puede basar este vencimiento en un número determinado de días desde que los objetos se vuelvan no actuales. Además del número de días, también puede indicar un número máximo de versiones no actuales que deben retenerse (entre 1 y 100). Este valor especifica cuántas versiones no actuales más recientes deben existir para que Amazon S3 pueda

realizar la acción asociada en una versión determinada. Para especificar la cantidad máxima de versiones no actuales, es necesario proporcionar un elemento `Filter`. Si no especifica un elemento `Filter`, Amazon S3 genera un error `InvalidRequest` cuando proporcione una cantidad máxima de versiones no actuales. Para obtener información sobre cómo usar la acción `NoncurrentVersionExpiration`, consulte the section called “Elementos para describir las acciones del ciclo de vida”.

- Para las acciones de Lifecycle, se comprueban dos tipos de metadatos de objetos: Bloqueo de objetos de S3 y un estado de replicación de `Pending`. Si Bloqueo de objetos de S3 está configurado en un objeto, Lifecycle no hace que venzan las versiones no actuales de ese objeto. Esta validación solo se comprueba cuando vence una clave con control de versiones o una versión no actual. Además, las claves con estados de replicación pendientes no están en transición ni vencen. Esta validación solo se aplica a las claves con control de versiones (tanto las versiones actuales como las no actuales).

Para obtener más información, consulte Usar el control de versiones en buckets de S3.

> **Important**
>
> Cuando tiene varias reglas en una configuración de S3 Lifecycle, un objeto puede reunir los requisitos para varias acciones de S3 Lifecycle realizadas el mismo día. En tales casos, Amazon S3 sigue estas reglas generales:
>
> - La eliminación permanente prevalece sobre la transición.
> - La transición prevalece sobre la creación de marcadores de eliminación.
> - Cuando un objeto es apto para una transición S3 Glacier Flexible Retrieval y una transición S3 Standard-IA (o una transición S3 One Zone-IA), Amazon S3 elige la transición S3 Glacier Flexible Retrieval.
>
> Para ver ejemplos, consulte Ejemplo 5: superposición de filtros, conflictos entre acciones de ciclo de vida y lo que hace Amazon S3 con buckets sin control de versiones.

Objetos existentes y nuevos

Cuando añade una configuración de ciclo de vida a un bucket, las reglas de configuración se aplican a los objetos existentes y a los objetos que añade posteriormente. Por ejemplo, si hoy agrega una regla de configuración de Lifecycle con una acción de vencimiento que causa que objetos con un

prefijo específico expiren 30 días después de la creación, Amazon S3 pondrá en cola de eliminación cualquier objeto existente con más de 30 días de antigüedad y que tenga el prefijo especificado.

> **Important**
>
> No puede usar una política de bucket para evitar eliminaciones ni transiciones mediante una regla de S3 Lifecycle. Por ejemplo, aunque la política de bucket deniegue todas las acciones a todas las entidades principales, la configuración de S3 Lifecycle seguirá funcionando con normalidad.

Cómo saber cuándo caducarán los objetos

Para conocer cuándo está previsto el vencimiento de un objeto, utilice las operaciones HeadObject o GetObject de la API. Estas operaciones de la API devuelven encabezados de respuesta que facilitan la fecha y la hora en que el objeto ya no se puede almacenar en la memoria caché.

> **Note**
>
> - Es posible que la fecha de vencimiento y la fecha en que Amazon S3 elimina un objeto estén desfasadas. No se le cobrará por el tiempo de almacenamiento asociado con un objeto que ha vencido.
> - Antes de actualizar, deshabilitar o eliminar las reglas del ciclo de vida, utilice las operaciones de la API de `LIST` (como ListObjectsV2, ListObjectVersions y ListMultipartUploads) o Inventario de Amazon S3 para comprobar que Amazon S3 haya realizado la transición y haya caducado los objetos aptos en función de sus casos de uso.

Cargo por duración mínima del almacenamiento

Si crea una regla de vencimiento de S3 Lifecycle para provocar el vencimiento de objetos que han estado en la clase de almacenamiento S3 Standard-IA o S3 One Zone-IA durante menos de 30 días, se le cobrarán 30 días. Si crea una regla de vencimiento del ciclo de vida para provocar el vencimiento de objetos que han estado en el almacenamiento S3 Glacier Flexible Retrieval durante menos de 90 días, se le cobrarán 90 días. Si crea una regla de vencimiento de Lifecycle para provocar el vencimiento de los objetos que han estado en el almacenamiento S3 Glacier Deep Archive durante menos de 180 días, se le cobrarán 180 días.

Para obtener más información, consulte Precios de Amazon S3.

Configuración de un ciclo de vida en un bucket

En esta sección, se explica cómo puede establecer una configuración de S3 Lifecycle en un bucket mediante la consola de Amazon S3, el AWS Command Line Interface (AWS CLI), los AWS SDK o la API de REST de Amazon S3. Para obtener información acerca de la configuración de S3 Lifecycle, consulte Administración del ciclo de vida del almacenamiento.

Puede usar las reglas de ciclo de vida para definir acciones que quiera que realice Amazon S3 durante el periodo de vida de un objeto (por ejemplo: la transición de objetos a otra clase de almacenamiento, su archivado o su eliminación tras transcurrir un periodo de tiempo especificado).

Antes de establecer una configuración del ciclo de vida, tenga en cuenta lo siguiente:

Retraso de propagación de configuración del ciclo de vida

Cuando añade una configuración de S3 Lifecycle a un bucket, por lo general se produce un cierto desfase antes de que una configuración de ciclo de vida nueva o actualizada se propague totalmente a todos los sistemas de Amazon S3. Habrá un retraso de algunos minutos antes de que la configuración entre en vigor completamente. Este retraso también se puede producir cuando elimina una configuración de S3 Lifecycle.

Retraso de caducidad o transición

Hay un retraso entre el momento en que se cumple la regla del ciclo de vida y el momento en que se completa su aplicación. Por ejemplo, supongamos que un conjunto de objetos caduca de acuerdo con una regla del ciclo de vida el 1 de enero. Aunque la regla de caducidad se haya cumplido el 1 de enero, es posible que Amazon S3 no elimine estos objetos hasta días o incluso semanas después. Este retraso se debe a que S3 Lifecycle pone los objetos en cola para transiciones o caducidades de forma asíncrona. Sin embargo, los cambios en la facturación suelen aplicarse en cuanto se cumple la regla de ciclo de vida, incluso aunque la acción no se haya completado. Para obtener más información, consulte Cambios en la facturación. Para supervisar el efecto de las actualizaciones realizadas por las reglas del ciclo de vida activo, consulte the section called “¿Cómo puedo supervisar las acciones que se llevan a cabo según mis reglas de ciclo de vida?”.

Deshabilitación o eliminación de las reglas del ciclo de vida

Cuando se deshabilitan o eliminan reglas del ciclo de vida después de cierto retraso, Amazon S3 deja de programar la eliminación o transición de nuevos objetos. Cualquier objeto que ya haya sido programado se desprograma y no se elimina ni se traslada.

> **Note**
>
> Antes de actualizar, deshabilitar o eliminar las reglas del ciclo de vida, utilice las operaciones de la API de `LIST` (como ListObjectsV2, ListObjectVersions y ListMultipartUploads) o Inventario de Amazon S3 para comprobar que Amazon S3 haya realizado la transición y haya caducado los objetos aptos en función de sus casos de uso. Si tiene problemas para actualizar, deshabilitar o eliminar las reglas del ciclo de vida, consulte Solucionar problemas de Amazon S3 Lifecycle.

Objetos existentes y nuevos

Cuando añade una configuración de ciclo de vida a un bucket, las reglas de configuración se aplican a los objetos existentes y a los objetos que añade posteriormente. Por ejemplo, si hoy agrega una regla de configuración de Lifecycle con una acción de vencimiento que causa que objetos con un prefijo específico expiren 30 días después de la creación, Amazon S3 pondrá en cola de eliminación cualquier objeto existente con más de 30 días de antigüedad y que tenga el prefijo especificado.

Supervisión del efecto de las reglas del ciclo de vida

Para supervisar el efecto de las actualizaciones realizadas por las reglas del ciclo de vida activo, consulte the section called “¿Cómo puedo supervisar las acciones que se llevan a cabo según mis reglas de ciclo de vida?”.

Cambios en la facturación

Puede que haya un desfase entre la fecha en que se cumplen las reglas de configuración de ciclo de vida y la fecha en que se realiza la acción para cumplir la regla. Sin embargo, los cambios en la facturación se producen en cuanto se cumple la regla de configuración de ciclo de vida, incluso aunque la acción todavía no se haya realizado.

Un ejemplo es cuando no se le cobra por el almacenamiento después del tiempo de vencimiento del objeto, incluso aunque el objeto no se elimine inmediatamente. De igual modo, se le cobran las tarifas de almacenamiento de S3 Glacier Flexible Retrieval en cuanto transcurre el tiempo de transición del objeto, incluso aunque el objeto no se traslade de inmediato a la clase de almacenamiento S3 Glacier Flexible Retrieval.

Sin embargo, las transiciones del ciclo de vida a la clase de almacenamiento de S3 Intelligent-Tiering son la excepción. Los cambios en la facturación no se producen hasta después de que el objeto haya pasado a la clase de almacenamiento de S3 Intelligent-Tiering.

Varias reglas o reglas incompatibles

Cuando tiene varias reglas en una configuración de S3 Lifecycle, un objeto puede reunir los requisitos para varias acciones de S3 Lifecycle realizadas el mismo día. En tales casos, Amazon S3 sigue estas reglas generales:

- La eliminación permanente prevalece sobre la transición.
- La transición prevalece sobre la creación de marcadores de eliminación.
- Cuando un objeto es apto para una transición S3 Glacier Flexible Retrieval y una transición S3 Standard-IA (o una transición S3 One Zone-IA), Amazon S3 elige la transición S3 Glacier Flexible Retrieval.

Para ver ejemplos, consulte Ejemplo 5: superposición de filtros, conflictos entre acciones de ciclo de vida y lo que hace Amazon S3 con buckets sin control de versiones.

Uso de la consola de S3

Puede definir una regla de Lifecycle para todos los objetos o para un subconjunto de objetos del bucket mediante un prefijo compartido (nombres de objetos que comienzan por una cadena común) o una etiqueta. En la regla de Lifecycle, puede definir acciones específicas para las versiones de objetos actualizadas y no actualizadas. Para más información, consulte los siguientes temas:

- Administración del ciclo de vida del almacenamiento
- Usar el control de versiones en buckets de S3

Para crear una regla de ciclo de vida

1. Inicie sesión en la AWS Management Console y abra la consola de Amazon S3 en https://console.aws.amazon.com/s3/.
2. En la lista Buckets, seleccione el nombre del bucket para el que desea crear una regla de ciclo de vida.
3. Seleccione la pestaña Management (Administración) y seleccione Create lifecycle rule (Crear regla de ciclo de vida).
4. En Lifecycle rule name (Nombre de regla de ciclo de vida), escriba un nombre para la regla.

   El nombre debe ser único dentro del bucket.
5. Elija el ámbito de la regla de ciclo de vida:

- Para aplicar esta regla de ciclo de vida a todos los objetos con un prefijo o etiqueta específicos, elija Limit the scope to specific prefixes or tags (Limitar el ámbito a prefijos o etiquetas específicos).
  - Para limitar el ámbito por prefijo, en Prefix (Prefijo), escriba el prefijo.
  - Para limitar el ámbito por etiqueta, seleccione Add tag (Agregar etiqueta) e introduzca la clave y el valor de la etiqueta.

  Para obtener más información acerca de los prefijos de nombre de objeto, consulte Creación de nombres de clave de objeto. Para obtener más información acerca de las etiquetas de objeto, consulte Categorización del almacenamiento mediante etiquetas.
- Para aplicar esta regla del ciclo de vida a todos los objetos del bucket, seleccione Esta regla se aplica a todos los objetos del bucket y, a continuación, elija Reconozco que esta regla se aplica a todos los objetos del bucket.

6. Para filtrar una regla por tamaño de objeto, puede seleccionar Especificar el tamaño mínimo de objeto, Especificar tamaño máximo de objeto o ambas opciones.

   - Cuando especifica un valor en Tamaño mínimo de objeto o Tamaño máximo de objeto, este debe ser superior a 0 bytes y hasta 5 TB. Puede indicar este valor en bytes, KB, MB o GB.
   - Cuando especifica ambos valores, el tamaño máximo de objeto debe ser superior al tamaño mínimo de objeto.

   > **Note**
   >
   > Los filtros Tamaño mínimo de objeto y Tamaño máximo de objeto excluyen los valores especificados. Por ejemplo, si configura un filtro para que caduquen los objetos con un tamaño mínimo de objeto de 128 KB, los objetos que tengan exactamente 128 KB no caducarán. En cambio, la regla solo se aplica a los objetos que tengan un tamaño superior a 128 KB.

7. En Lifecycle rule actions (Acciones de regla de ciclo de vida), elija las acciones que desea que realice la regla de ciclo de vida:

   - Realizar la transición de versiones de objetos actuales entre clases de almacenamiento
   - Realizar la transición de versiones de objetos anteriores entre clases de almacenamiento
   - Caducar las versiones de objetos actuales

> **Note**
>
> En el caso de los buckets que no tengan habilitado el control de versiones de S3, la caducidad de las versiones actuales hace que Amazon S3 elimine los objetos de forma permanente. Para obtener más información, consulte the section called "Acciones de ciclo de vida y estado de control de versiones del bucket".

- Eliminar permanentemente versiones de objetos anteriores
- Eliminar marcadores de eliminación caducados o cargas multiparte incompletas

Dependiendo de las acciones que elija, aparecerán diferentes opciones.

8. Para realizar la transición de versiones de objetos actuales entre clases de almacenamiento, en Transition current versions of objects between storage classes (Realizar la transición de versiones de objetos actuales entre clases de almacenamiento):

   a. En Transiciones de clase de almacenamiento, seleccione la clase de almacenamiento a la que quiera realizar la transición. Para ver una lista de posibles transiciones, consulte the section called "Transiciones del ciclo de vida admitidas". Puede elegir entre las siguientes clases de almacenamiento:

      - S3 Standard-IA
      - S3 Intelligent-Tiering
      - S3 One Zone-IA
      - S3 Glacier Flexible Retrieval
      - S3 Glacier Deep Archive

   b. En Days after object creation (Días después de la creación del objeto), introduzca el número de días posteriores a la creación del objeto en los que quiera realizar la transición.

   Para obtener más información acerca de las clases de almacenamiento, consulte Uso de las clases de almacenamiento de Amazon S3. Puede definir transiciones para versiones de objetos actuales o anteriores, o tanto para las actuales como para las anteriores. El control de versiones le permite mantener varias versiones de un objeto en un bucket. Para obtener más información sobre el control de versiones, consulte Uso de la consola de S3.

> **Important**
>
> Si elige la clase de almacenamiento S3 Glacier Flexible Retrieval o Glacier Deep Archive, los objetos permanecen en Amazon S3. No puede acceder a ellos directamente a través del servicio independiente de Amazon S3 Glacier. Para obtener más información, consulte Transición de objetos con Amazon S3 Lifecycle.

9. Para realizar la transición de versiones de objetos no actuales entre clases de almacenamiento, en Transition noncurrent versions of objects between storage classes (Realizar la transición de versiones de objetos no actuales entre clases de almacenamiento):

   a. En Transiciones de clase de almacenamiento, seleccione la clase de almacenamiento a la que quiera realizar la transición. Para ver una lista de posibles transiciones, consulte the section called “Transiciones del ciclo de vida admitidas”. Puede elegir entre las siguientes clases de almacenamiento:

      - S3 Standard-IA
      - S3 Intelligent-Tiering
      - S3 One Zone-IA
      - S3 Glacier Flexible Retrieval
      - S3 Glacier Deep Archive

   b. En Days after object becomes non-current (Días después de que el objeto se vuelve no actual), introduzca el número de días posteriores a la creación del objeto en los que quiera realizar la transición.

10. Para hacer caducar versiones de objetos actuales, en Expire current versions of objects (Hacer caducar versiones de objetos actuales), en Number of days after object creation (Número de días después de la creación del objeto), ingrese el número de días.

> **Important**
>
> En un bucket sin control de versiones, la acción de vencimiento da como resultado que Amazon S3 elimine de forma permanente el objeto. Para obtener más información sobre las acciones del ciclo de vida, consulte Elementos para describir las acciones del ciclo de vida.

11. Para eliminar de manera permanente versiones anteriores de objetos, en Permanently delete noncurrent versions of objects (Eliminar de manera permanente versiones no actuales de objetos), en Days after objects become noncurrent (Días después de que los objetos se vuelven no actuales), escriba el número de días. Puede especificar opcionalmente el número de versiones más recientes que desea retener introduciendo un valor en Number of newer versions to retain (Número de versiones más recientes que se deben retener).
12. Bajo Delete expired markers or incomplete multipart uploads (Eliminar marcadores caducados o cargas multiparte incompletas), seleccione Delete expired object delete markers (Eliminar marcadores de eliminación de objetos caducados) y Delete incomplete multipart uploads (Eliminar cargas multiparte incompletas). A continuación, escriba el número de días que han de transcurrir entre el inicio de la carga multiparte y el momento en que quiera finalizarla y limpiar las cargas incompletas.

    Para obtener más información acerca de las cargas multipartes, consulte Carga y copia de objetos con la carga multiparte.
13. Elija Create rule (Crear regla).

    Si la regla no contiene ningún error, Amazon S3 la habilita y se puede ver en la ficha Management (Administración) en Lifecycle rules (Reglas del ciclo de vida).

Para obtener información sobre el uso de plantillas de AWS CloudFormation y algunos ejemplos, consulte Trabajo con plantillas de AWS CloudFormation y AWS::S3::Bucket en la Guía del usuario de AWS CloudFormation.

Mediante AWS CLI

Puede utilizar los siguientes comandos de la AWS CLI para administrar configuraciones de ciclo de vida de S3:

- `put-bucket-lifecycle-configuration`
- `get-bucket-lifecycle-configuration`
- `delete-bucket-lifecycle`

Para obtener instrucciones acerca de cómo configurar la AWS CLI, consulte Desarrollo con Amazon S3 mediante la AWS CLI.

La configuración de Amazon S3 Lifecycle es un archivo XML. Pero cuando se usa la AWS CLI no se puede especificar el formato XML. Debe especificar el formato JSON en su lugar. A continuación, se indican ejemplos de configuraciones de Lifecycle de XML y las configuraciones de JSON equivalentes que puede especificar en un comando de AWS CLI.

Considere la siguiente configuración de S3 Lifecycle de ejemplo:

Example Ejemplo 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 Ejemplo 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
            }
        }
    ]
}
```

Puede probar la `put-bucket-lifecycle-configuration` de la siguiente manera.

Para probar la configuración

1. Guarde la configuración de Lifecycle de JSON en un archivo (por ejemplo, *`lifecycle.json`*).
2. Ejecute el siguiente comando de la AWS CLI para establecer la configuración del ciclo de vida en su bucket. Reemplace los *`user input placeholders`* con su propia información.

```
$ aws s3api put-bucket-lifecycle-configuration  \
--bucket amzn-s3-demo-bucket  \
--lifecycle-configuration file://lifecycle.json
```

3. Para verificar, recupere la configuración de S3 Lifecycle con el comando `get-bucket-lifecycle-configuration` de la AWS CLI de la siguiente manera:

```
$ aws s3api get-bucket-lifecycle-configuration  \
--bucket amzn-s3-demo-bucket
```

4. Para eliminar la configuración de S3 Lifecycle, utilice el comando `delete-bucket-lifecycle` de la AWS CLI de la siguiente manera.

```
aws s3api delete-bucket-lifecycle \
--bucket amzn-s3-demo-bucket
```

Uso de los AWS SDK

Java

Puede utilizar AWS SDK for Java para administrar la configuración del ciclo de vida de S3 de un bucket. Para obtener más información acerca de cómo administrar la configuración de S3 Lifecycle, consulte Administración del ciclo de vida del almacenamiento.

> **Note**
>
> Cuando añade una configuración de S3 Lifecycle a un bucket, Amazon S3 reemplaza la configuración de Lifecycle actual, si la hubiera. Para actualizar una configuración, debe recuperarla, hacer los cambios deseados y luego añadir la configuración revisada al bucket.

El siguiente ejemplo muestra cómo usar AWS SDK for Java para añadir, actualizar y eliminar una configuración de ciclo de vida de un bucket. En el ejemplo se realiza lo siguiente:

- Añade una configuración de ciclo de vida a un bucket.
- Recupera la configuración de ciclo de vida y la actualiza añadiendo otra regla.
- Añade la configuración de Lifecycle modificada al bucket. Amazon S3 reemplaza la configuración existente.
- Recupera la configuración de nuevo y verifica que tiene el número correcto de reglas imprimiendo el número de reglas.

- Elimina la configuración de nuevo y verifica que se ha eliminado intentando volver a recuperarla.

Para obtener instrucciones sobre cómo crear y probar una muestra funcional, consulte Introducción en la Guía del desarrollador de AWS SDK for Java.

```
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

Puede utilizar AWS SDK for .NET para administrar la configuración del ciclo de vida de S3 en un bucket. Para obtener más información acerca de cómo administrar la configuración del ciclo de vida, consulte Administración del ciclo de vida del almacenamiento.

> **Note**
>
> Cuando añade una configuración de ciclo de vida, Amazon S3 reemplaza la configuración existente en el bucket especificado. Para actualizar una configuración, primero debe recuperar la configuración de ciclo de vida existente, hacer los cambios y luego añadir la configuración de ciclo de vida revisada al bucket.

El siguiente ejemplo muestra cómo usar AWS SDK for .NET para añadir, actualizar y eliminar una configuración de ciclo de vida de un bucket. Este ejemplo de código hace lo siguiente:

- Añade una configuración de ciclo de vida a un bucket.
- Recupera la configuración de ciclo de vida y la actualiza añadiendo otra regla.
- Añade la configuración de Lifecycle modificada al bucket. Amazon S3 reemplaza la configuración de ciclo de vida existente.
- Recupera la configuración de nuevo y la verifica imprimiendo el número de reglas en la configuración.
- Elimina la configuración del ciclo de vida y verifica la eliminación.

Para obtener información acerca de cómo configurar y ejecutar ejemplos de código, consulte Introducción al SDK de AWS para .NET en la Guía para desarrolladores del SDK de AWS para .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

Puede utilizar AWS SDK for Ruby para administrar la configuración del ciclo de vida de S3 en un bucket con la clase AWS::S3::BucketLifecycleConfiguration. Para obtener más información acerca de cómo administrar la configuración del ciclo de vida, consulte Administración del ciclo de vida del almacenamiento.

Uso de la API de REST

En las secciones siguientes de la referencia de la API de Amazon Simple Storage Service se describe la API de REST relacionada con la configuración de S3 Lifecycle.

- PutBucketLifecycleConfiguration
- GetBucketLifecycleConfiguration
- DeleteBucketLifecycle

Solución de problemas del ciclo de vida

Para ver problemas habituales que pueden producirse al trabajar con S3 Lifecycle, consulte the section called “Solucionar problemas de ciclo de vida”.

Configuraciones del ciclo de vida y otras configuraciones del bucket

Además de las configuraciones de S3 Lifecycle, puede asociar otras configuraciones con el bucket. En esta sección se explica cómo la configuración de S3 Lifecycle se relaciona con otras configuraciones del bucket.

Ciclos de vida y control de versiones

Puede añadir configuraciones de S3 Lifecycle a buckets sin control de versiones y buckets con control de versiones habilitado. Para obtener más información, consulte Usar el control de versiones en buckets de S3.

Un bucket habilitado para el control de versiones mantiene una versión actual del objeto y cero o más versiones no actuales del objeto. Puede definir diferentes reglas de ciclo de vida para las versiones actuales y no actuales del objeto.

Para obtener más información, consulte Elementos de configuración del ciclo de vida.

> **Important**
>
> Cuando tiene varias reglas en una configuración de S3 Lifecycle, un objeto puede reunir los requisitos para varias acciones de S3 Lifecycle realizadas el mismo día. En tales casos, Amazon S3 sigue estas reglas generales:
>
> - La eliminación permanente prevalece sobre la transición.
> - La transición prevalece sobre la creación de marcadores de eliminación.
> - Cuando un objeto es elegible para una transición S3 Glacier Flexible Retrieval y S3 Standard-IA (o S3 One Zone-IA), Amazon S3 elige la transición S3 Glacier Flexible Retrieval.
>
> Para ver ejemplos, consulte Ejemplo 5: superposición de filtros, conflictos entre acciones de ciclo de vida y lo que hace Amazon S3 con buckets sin control de versiones.

Configuración del ciclo de vida en buckets habilitados para la MFA

No se admite la configuración del ciclo de vida en buckets habilitados para autenticación multifactor (MFA).

Ciclos de vida y registros

Las acciones de ciclo de vida de Amazon S3 no se capturan mediante el registro de objeto de AWS CloudTrail. CloudTrail captura las solicitudes de la API realizadas a puntos de enlace externos de Amazon S3, mientras que las acciones de S3 Lifecycle se realizan mediante puntos de enlace internos de Amazon S3. Los registros de acceso a servidores de Amazon S3 se pueden habilitar en un bucket de S3 para capturar acciones relacionadas con S3 Lifecycle como la transición de objetos a otra clase de almacenamiento y la caducidad del objeto, lo que provoca una eliminación permanente o eliminación lógica. Para obtener más información, consulte the section called “Registro de acceso al servidor”.

Si tiene el registro habilitado en su bucket, los registros de acceso del servidor de Amazon S3 notifican los resultados de las siguientes operaciones.

Registro de operaciones	Descripción
`S3.EXPIRE.OBJECT`	Amazon S3 elimina de forma permanente el objeto por la acción de vencimiento de su ciclo de vida.
`S3.CREATE.DELETEMARKER`	Amazon S3 elimina de forma lógica la versión actual y añade el marcador de eliminación en un bucket que tenga el control de versiones habilitado.
`S3.TRANSITION_SIA.OBJECT`	Amazon S3 pasa el objeto a la clase de almacenamiento S3 Standard-IA.
`S3.TRANSITION_ZIA.OBJECT`	Amazon S3 pasa el objeto a la clase de almacenamiento S3 One Zone-IA.
`S3.TRANSITION_INT.OBJECT`	Amazon S3 pasa el objeto a la clase de almacenamiento S3 Intelligent-Tiering.
`S3.TRANSITION_GIR.OBJECT`	Amazon S3 inicia la transición del objeto a la clase de almacenamiento S3 Glacier Instant Retrieval.
`S3.TRANSITION.OBJECT`	Amazon S3 inicia la transición del objeto a la clase de almacenamiento S3 Glacier Flexible Retrieval.
`S3.TRANSITION_GDA.OBJECT`	Amazon S3 inicia la transición del objeto a la clase de almacenamiento S3 Glacier Deep Archive.
`S3.DELETE.UPLOAD`	Amazon S3 anula una carga multiparte incompleta.

> **Note**
>
> Los registros de acceso del servidor de Amazon S3 se entregan habitualmente en la medida de lo posible y no se pueden utilizar para completar la contabilidad de todas las solicitudes de Amazon S3.

Solución de problemas del ciclo de vida

Para obtener más información acerca de cómo solucionar problemas habituales con Ciclo de vida de S3, consulte Solucionar problemas de Amazon S3 Lifecycle.

Más información

- Elementos de configuración del ciclo de vida
- Transición a las clases de almacenamiento S3 Glacier Flexible Retrieval y S3 Glacier Deep Archive (archivo de objetos)
- Configuración de un ciclo de vida en un bucket

Configurar notificaciones de eventos de Lifecycle

Puede configurar una notificación de evento de Amazon S3 para recibir notificación cuando Amazon S3 elimina un objeto o lo pasa a otra clase de almacenamiento de Amazon S3 siguiendo una regla de S3 Lifecycle.

Mediante el uso de los tipos de eventos `LifecycleExpiration`, puede recibir notificaciones cada vez que Amazon S3 elimina un objeto en función de la configuración de S3 Lifecycle. El tipo de evento `s3:LifecycleExpiration:Delete` le notifica cuando se elimina un objeto de un bucket sin control de versiones. También le notifica cuando la versión de un objeto se elimina de forma permanente mediante una configuración de S3 Lifecycle. El tipo de evento `s3:LifecycleExpiration:DeleteMarkerCreated` le notifica cuando S3 Lifecycle crea un marcador de eliminación cuando se elimina la versión actual de un objeto del bucket con control de versiones. Para obtener más información, consulte Delete object version (Eliminar versión de objeto).

Mediante el uso del tipo de evento `s3:LifecycleTransition`, puede recibir notificación cuando un objeto se transfiera a otra clase de almacenamiento de Amazon S3 mediante una configuración de S3 Lifecycle.

Amazon S3 puede publicar notificaciones de eventos en un tema de Amazon Simple Notification Service (Amazon SNS), una cola de Amazon Simple Queue Service (Amazon SQS) o una función de AWS Lambda. Para obtener más información, consulte Notificaciones de eventos de Amazon S3.

Para obtener instrucciones sobre cómo configurar notificaciones de eventos de Amazon S3, consulte Enabling event notifications (Habilitación de notificaciones de eventos).

El siguiente mensaje es un ejemplo de un mensaje que Amazon S3 envía para publicar un evento de `s3:LifecycleExpiration:Delete`. Para obtener más información, consulte Estructura de mensajes de evento.

```
{
   "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":"amzn-s3-demo-bucket",
               "ownerIdentity":{
                  "principalId":"A3NL1KOZZKExample"
               },
               "arn":"arn:aws:s3:::amzn-s3-demo-bucket"
            },
            "object":{
               "key":"expiration/delete",
               "sequencer":"0055AED6DCD90281E5",
```

```
            }
         }
      }
   ]
}
```

Mensajes que Amazon S3 envía para publicar un evento `s3:LifecycleTransition` que también incluye la siguiente información.

```
"lifecycleEventData":{
    "transitionEventData": {
        "destinationStorageClass": the destination storage class for the object
    }
}
```

Elementos de configuración del ciclo de vida

Temas

- Elemento ID
- Elemento Status
- Elemento Filter
- Elementos para describir las acciones del ciclo de vida

La configuración de Amazon S3 Lifecycle se especifica mediante un XML que consiste en una o varias reglas de Lifecycle.

```
<LifecycleConfiguration>
    <Rule>
         ...
    </Rule>
    <Rule>
         ...
    </Rule>
</LifecycleConfiguration>
```

Cada regla consta de los elementos siguientes:

- Metadatos de la regla, que incluyen un ID de regla y un estado que indica si esta está activada o desactivada. Si una regla está desactivada, Amazon S3 no realiza ninguna de las acciones especificadas en ella.
- Un filtro que identifica los objetos a los que se aplica la regla. Puede especificar un filtro utilizando un tamaño de objeto, un prefijo de clave de objeto, una o varias etiquetas de objeto o una combinación de filtros.
- Una o varias acciones de transición o vencimiento con una fecha o un periodo de tiempo en la vida del objeto en el que quiera que Amazon S3 realice la acción especificada.

En las siguientes secciones se describen los elementos XML de una configuración de S3 Lifecycle. Para ver configuraciones de ejemplo, consulte Ejemplos de configuración de S3 Lifecycle.

Elemento ID

Una configuración de S3 Lifecycle puede tener hasta 1000 reglas. Este límite no se puede ajustar. El elemento `<ID>` identifica unívocamente una regla. La longitud del ID está limitada a 255 caracteres.

Elemento Status

El valor del elemento `<Status>` puede ser `Enabled` o `Disabled`. Si una regla está desactivada, Amazon S3 no realiza ninguna de las acciones definidas en ella.

Elemento Filter

Una regla de Lifecycle se puede aplicar a todos los objetos o a un subconjunto de los objetos de un bucket, en función del elemento `<Filter>` que especifique en la regla de Lifecycle.

Puede filtrar los objetos por prefijo de clave, etiquetas de objeto o una combinación de ambos (en cuyo caso, Amazon S3 utilizará un operador lógico `AND` para combinar los filtros). Considere los siguientes ejemplos:

- Especificación de un filtro mediante prefijos de clave: en este ejemplo se muestra una regla de S3 Lifecycle que se aplica a un subconjunto de objetos en función del prefijo del nombre de la clave (`logs/`). Por ejemplo, la regla de Lifecycle se aplica a los objetos `logs/mylog.txt`, `logs/temp1.txt` y `logs/test.txt`. La regla no se aplica al objeto `example.jpg`.

```
<LifecycleConfiguration>
    <Rule>
        <Filter>
```

```
            <Prefix>logs/</Prefix>
        </Filter>
        transition/expiration actions
         ...
    </Rule>
    ...
</LifecycleConfiguration>
```

Si quiere aplicar una acción de ciclo de vida a un subconjunto de objetos en función de diferentes prefijos de nombres de clave, especifique reglas por separado. En cada regla, especifique un filtro basado en prefijos. Por ejemplo, para describir una acción de Lifecycle para objetos con los prefijos de clave `projectA/` y `projectB/`, debe especificar dos reglas como se muestra a continuación:

```
<LifecycleConfiguration>
    <Rule>
        <Filter>
           <Prefix>projectA/</Prefix>
        </Filter>
        transition/expiration actions
         ...
    </Rule>

    <Rule>
        <Filter>
           <Prefix>projectB/</Prefix>
        </Filter>
        transition/expiration actions
         ...
    </Rule>
</LifecycleConfiguration>
```

Para obtener más información sobre las claves de objetos, consulte Creación de nombres de clave de objeto.

- Especificación de un filtro mediante etiquetas de objetos: en el siguiente ejemplo, la regla de Lifesycle especifica un filtro basado en una etiqueta (*key*) y valor (*value*). La regla se aplica solo a un subconjunto de objetos que tengan la etiqueta específica.

```
<LifecycleConfiguration>
    <Rule>
```

```
        <Filter>
           <Tag>
              <Key>key</Key>
              <Value>value</Value>
           </Tag>
        </Filter>
        transition/expiration actions
        ...
    </Rule>
</LifecycleConfiguration>
```

Puede especificar un filtro sobre la base de varias etiquetas. Debe envolver las etiquetas con el elemento <And>, tal y como se muestra en el siguiente ejemplo. La regla indica a Amazon S3 que debe realizar acciones de ciclo de vida en objetos con dos etiquetas (con la clave y el valor específicos de la etiqueta).

```
<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 regla de Lifecycle se aplica solo a un subconjunto de objetos que tengan la etiqueta específica. Amazon S3 realiza una operación lógica AND. Tenga en cuenta lo siguiente:

- Cada etiqueta debe coincidir exactamente tanto con la clave como con el valor. Si especifica solo un elemento <Key> y no especifica ningún elemento <Value>, la regla se aplicará únicamente a los objetos que coincidan con la clave de la etiqueta y que no tengan un valor especificado.

- La regla se aplica al subconjunto de objetos que tienen todas las etiquetas especificadas en la regla. Si un objeto tiene etiquetas adicionales especificadas, se seguirá aplicando la regla.

> **Note**
>
> Al especificar varias etiquetas en un filtro, cada clave de etiqueta ha de ser exclusiva.

- Especificación de un filtro utilizando tanto el prefijo como una o varias etiquetas de objetos: en una regla de Lifecycle, puede especificar un filtro basado tanto en el prefijo de clave como en una o varias etiquetas. Una vez más, debe envolver todos estos elementos de filtro con el elemento <And> como se muestra a continuación:

```
<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 combina estos filtros mediante el operador lógico AND. Es decir, la regla se aplica al subconjunto de objetos con el prefijo de clave específico y las etiquetas específicas. Un filtro solo puede tener un prefijo y ninguna, una o varias etiquetas.

- Puede especificar un filtro vacío, en cuyo caso, la regla se aplicará a todos los objetos del bucket.

```
<LifecycleConfiguration>
    <Rule>
```

```
        <Filter>
        </Filter>
        <Status>Enabled</Status>
        transition/expiration actions
    </Rule>
</LifecycleConfiguration>
```

- Para filtrar una regla por tamaño de objeto, puede especificar un tamaño mínimo (`ObjectSizeGreaterThan`) o un tamaño máximo (`ObjectSizeLessThan`), o bien puede especificar un rango de tamaños de objeto.

  Los valores de tamaño de objeto están expresados en bytes. El tamaño máximo del filtro es de 5 TB. Algunas clases de almacenamiento tienen limitaciones mínimas de tamaño de objeto. Para obtener más información, consulte Comparación de las clases de almacenamiento de Amazon S3.

```
<LifecycleConfiguration>
    <Rule>
        <Filter>
            <ObjectSizeGreaterThan>500</ObjectSizeGreaterThan>
        </Filter>
        <Status>Enabled</Status>
        transition/expiration actions
    </Rule>
</LifecycleConfiguration>
```

> **Note**
>
> Los filtros `ObjectSizeGreaterThan` y `ObjectSizeLessThan` excluyen los valores especificados. Por ejemplo, si establece objetos con un tamaño que abarca desde los 128 KB hasta los 1024 KB para pasar de la clase de almacenamiento S3 Standard a la clase de almacenamiento S3 Standard-IA, los objetos que tienen un tamaño exacto de 1024 KB y de 128 KB no pasarán a S3 Standard-IA. En cambio, la regla se aplicará únicamente a los objetos que tengan un tamaño superior a 128 KB e inferior a 1024 KB.

Si especifica un rango de tamaño de objeto, el número entero `ObjectSizeGreaterThan` debe ser menor que valor `ObjectSizeLessThan`. Cuando utilice más de un filtro, debe envolver los filtros en un elemento `<And>`. En el siguiente ejemplo, se muestra cómo especificar objetos en un rango de entre 500 y 64 000 bytes.

```
<LifecycleConfiguration>
    <Rule>
        <Filter>
            <And>
                <Prefix>key-prefix</Prefix>
                <ObjectSizeGreaterThan>500</ObjectSizeGreaterThan>
                <ObjectSizeLessThan>64000</ObjectSizeLessThan>
            </And>
        </Filter>
        <Status>Enabled</Status>
        transition/expiration actions
    </Rule>
</LifecycleConfiguration>
```

Elementos para describir las acciones del ciclo de vida

Puede ordenar a Amazon S3 que realice acciones específicas durante la vida útil de un objeto especificando una o varias de las siguientes acciones predefinidas en una regla de S3 Lifecycle. El efecto de estas acciones depende del estado del control de versiones del bucket.

- Elemento de acción **`Transition`**: puede especificar la acción `Transition` para pasar objetos de una clase de almacenamiento a otra. Para obtener más información acerca de las transiciones de objetos, consulte Transiciones admitidas y limitaciones relacionadas. Cuando se llega a una fecha o periodo de tiempo específico en la vida útil del objeto, Amazon S3 lleva a cabo la transición.

  Para un bucket con control de versiones (un bucket con el control de versiones habilitado o suspendido), la acción `Transition` se aplica a la versión del objeto actual. Para administrar las versiones no actuales, Amazon S3 define la acción `NoncurrentVersionTransition` (que se describe más adelante en este tema).

- Elemento de acción **`Expiration`**: la acción `Expiration` hace que caduquen objetos identificados por la regla y se aplica a los objetos elegibles de cualquiera de las clases de almacenamiento de Amazon S3. Para obtener más información acerca de las clases de almacenamiento, consulte Uso de las clases de almacenamiento de Amazon S3. Amazon S3 hace que todos los objetos vencidos dejen de estar disponibles. La eliminación permanente de los objetos dependerá del estado del bucket en relación con el control de versiones.

- Bucket sin control de versiones: la acción `Expiration` da como resultado la eliminación permanente del objeto por parte de Amazon S3.
- Bucket con control de versiones: para un bucket con control de versiones (un bucket con el control de versiones habilitado o suspendido), hay varias consideraciones que se deben tener en cuenta en relación con cómo administra Amazon S3 la acción `Expiration`. En el caso de los buckets con el control de versiones activado o suspendido, se aplica lo siguiente:
  - La acción `Expiration` solo se aplica a la versión actual (no tiene ningún impacto sobre las versiones del objeto no actuales).
  - Amazon S3 no realiza ninguna acción si hay una o varias versiones del objeto y el marcador de eliminación es la versión actual.
  - Si la versión actual del objeto es la única versión del objeto y, además, es un marcador de eliminación (también denominado marcador de eliminación de objeto vencido, en el que todas las versiones del objeto se han eliminado y solo queda un marcador de eliminación), Amazon S3 elimina el marcador de eliminación del objeto vencido. También puede usar la acción de vencimiento para ordenar a Amazon S3 que elimine los marcadores de eliminación de objeto vencidos. Para ver un ejemplo, consulte Ejemplo 7: eliminar marcadores de eliminación de objetos que vencieron.

  Para obtener más información, consulte Usar el control de versiones en buckets de S3.

  Tenga en cuenta también lo siguiente al configurar Amazon S3 para administrar el vencimiento:

  - Bucket con control de versiones habilitado

    Si la versión actual del objeto no es un marcador de eliminación, Amazon S3 añade un marcador de eliminación con un ID de versión exclusivo. Esto convierte la versión actual en no actual, y el marcador de eliminación en la versión actual.
  - Bucket con control de versiones suspendido

    En un bucket con el control de versiones suspendido, la acción de vencimiento provoca que Amazon S3 cree un marcador de eliminación con un ID de versión `null`. Este marcador de eliminación sustituye cualquier versión del objeto con un ID de versión null en la jerarquía de versiones, lo que elimina el objeto de manera eficaz.

Además, Amazon S3 facilita las siguientes acciones que puede utilizar para administrar las versiones de objetos no actuales en un bucket con control de versiones (en buckets con control de versiones activado o suspendido).

- Elemento de acción **`NoncurrentVersionTransition`**: utilice esta acción para especificar cuándo Amazon S3 debe realizar la transición de los objetos a la clase de almacenamiento especificada. Puede basar este vencimiento en un número determinado de días desde que los objetos se vuelvan no actuales. Además del número de días, también puede indicar un número máximo de versiones no actuales que deben retenerse (entre 1 y 100). Este valor especifica cuántas versiones no actuales más recientes deben existir para que Amazon S3 pueda realizar la acción asociada en una versión determinada. Amazon S3 realizará la transición de cualquier versión no actual adicional más allá del número especificado para conservar.

  Para especificar la cantidad máxima de versiones no actuales, es necesario proporcionar un elemento `Filter`. Si no especifica un elemento `Filter`, Amazon S3 genera un error `InvalidRequest` cuando proporcione una cantidad máxima de versiones no actuales.

  Para obtener más información acerca de las transiciones de objetos, consulte Transiciones admitidas y limitaciones relacionadas. Para obtener más información sobre cómo Amazon S3 calcula la fecha en la que especifica el número de días en la acción `NoncurrentVersionTransition`, consulte Reglas de ciclo de vida: basadas en la edad de un objeto.
- Elemento de acción **`NoncurrentVersionExpiration`**: utilice esta acción para especificar a Amazon S3 que elimine de forma permanente versiones no actuales de objetos. Estos objetos eliminados no se pueden recuperar. Puede basar este vencimiento en un número determinado de días desde que los objetos se vuelvan no actuales. Además del número de días, también puede indicar un número máximo de versiones no actuales que deben retenerse (entre 1 y 100). Este valor especifica cuántas versiones no actuales más recientes deben existir para que Amazon S3 pueda realizar la acción asociada en una versión determinada. Amazon S3 eliminará de forma definitiva cualquier versión no actual adicional que supere el número especificado para conservar.

  Para especificar la cantidad máxima de versiones no actuales, es necesario proporcionar un elemento `Filter`. Si no especifica un elemento `Filter`, Amazon S3 genera un error `InvalidRequest` cuando proporcione una cantidad máxima de versiones no actuales.

  La eliminación retrasada de objetos no actuales puede resultar útil si necesita corregir eliminaciones o sobrescrituras accidentales. Por ejemplo, puede configurar una regla de vencimiento para eliminar las versiones no actuales después de cinco días desde que dejan de ser actuales. Por ejemplo, suponga que el 01/01/2014, a las 10:30 h UTC, creó un objeto denominado `photo.gif` (ID de versión 111111). El 02/01/2014, a las 11:30 h UTC, eliminó por accidente `photo.gif` (ID de versión 111111), lo que crea un marcador de eliminación con un nuevo ID de versión (por ejemplo: ID de versión 4857693). A partir de este momento tendrá cinco días para

recuperar la versión original de `photo.gif` (ID de versión 111111) antes de que la eliminación sea permanente. El 08/01/2014, a las 00:00 h UTC, se ejecutará la regla de Lifecycle para el vencimiento y se eliminará permanentemente `photo.gif` (ID de versión 111111), cinco días después de que la versión dejase de ser actual.

Para obtener más información sobre cómo Amazon S3 calcula la fecha en la que especifica el número de días en una acción `NoncurrentVersionExpiration`, consulte Reglas de ciclo de vida: basadas en la edad de un objeto.

> **Note**
>
> Las configuraciones del ciclo de vida de vencimiento de los objetos no eliminan las cargas multiparte incompletas. Para eliminar las cargas multiparte incompletas, deberá usar la acción de configuración Lifecycle `AbortIncompleteMultipartUpload` que se describe más adelante en esta sección.

Además de las acciones de transición y vencimiento, puede usar las siguientes acciones de configuración de Lifecycle para ordenar a Amazon S3 que detenga las cargas multiparte incompletas o eliminar los marcadores de eliminación de objetos que vencieron.

- Elemento de acción **`AbortIncompleteMultipartUpload`**: use este elemento para establecer un tiempo máximo (en días) que quiera permitir que las cargas multiparte sigan estando en curso. Si las cargas multiparte aplicables (determinadas por el nombre de clave `prefix` especificado en la regla de Lifecycle) no se completan satisfactoriamente en el período de tiempo predefinido, Amazon S3 detiene las cargas multiparte incompletas. Para obtener más información, consulte Anulación de la carga multiparte.

  > **Note**
  >
  > No puede especificar esta acción de Lifecycle en una regla que tenga un filtro que use etiquetas de objetos.

- Elemento de acción **`ExpiredObjectDeleteMarker`**: en un bucket con control de versiones habilitado, un marcador de eliminación sin versiones no actuales se denomina marcador de eliminación del objeto vencido. Puede usar esta acción de Lifecycle para ordenar a S3 que elimine los marcadores de eliminación del objetos vencidos. Para ver un ejemplo, consulte Ejemplo 7: eliminar marcadores de eliminación de objetos que vencieron.

> **Note**
>
> No puede especificar esta acción de Lifecycle en una regla que tenga un filtro que use etiquetas de objetos.

Cómo calcula Amazon S3, el tiempo que un objeto lleva siendo no actual

Es posible tener varias versiones de un objeto en un bucket con control de versiones habilitado. Siempre habrá una versión actual y ninguna o varias versiones no actuales. Cada vez que carga un objeto, se conserva la versión actual como versión no actual, y la versión nueva, su sucesora, será la nueva versión actual. Para determinar el número de días que un objeto lleva siendo no actual, Amazon S3 analiza el momento en el que se creó su sucesor. Amazon S3 usa el número de días desde la creación de su sucesor como número de días durante los que un objeto no es actual.

> **Restauración de versiones anteriores de un objeto al utilizar configuraciones de S3 Lifecycle**
>
> Como se explica en Restaurar versiones anteriores, puede usar cualquiera de los siguientes dos métodos para recuperar versiones anteriores de un objeto:
>
> - Método 1: copiar una versión no actual del objeto en el mismo bucket. El objeto copiado se convierte en la versión actual del mismo, y se conservan todas las versiones del objeto.
> - Método 2: eliminar permanentemente la versión actual del objeto. Al eliminar la versión actual del objeto, en efecto, estará convirtiendo la versión no actual en la versión actual del mismo.
>
> Cuando use las reglas de configuración de S3 Lifecycle en buckets con control de versiones, nuestra recomendación es que use el Método 1.
> S3 Lifecycle funciona bajo un modelo de coherencia final. Una versión actual eliminada permanentemente podría no desaparecer hasta que se propaguen los cambios a todos los sistemas de Amazon S3. (Por lo tanto, es posible que Amazon S3 desconozca temporalmente esta eliminación). Mientras tanto, la regla del ciclo de vida que haya configurado para hacer que venzan los objetos no actuales podría eliminar permanentemente los objetos no actuales, incluido el que quiera restaurar. Por ello, copiar la versión antigua, como se recomienda en el Método 1, es la alternativa más segura.

Acciones de ciclo de vida y estado de control de versiones del bucket

Reglas de ciclo de vida: basadas en la edad de un objeto

Puede especificar un período de tiempo, en número de días a partir de la creación (o modificación) del objeto, en el que Amazon S3 puede llevar a cabo la acción especificada.

Al especificar el número de días en las acciones `Transition` y `Expiration` de una configuración de S3 Lifecycle, tenga en cuenta lo siguiente:

- El valor que especifique es el número de días desde la creación del objeto tras los que ocurrirá la acción.
- Amazon S3 calcula el tiempo agregando el número de días especificados en la regla al momento de creación del objeto y redondeando el tiempo resultante a la medianoche del día siguiente, hora UTC. Por ejemplo, si un objeto se creó el 15/01/2014, a las 10:30 h, UTC, y la cantidad de días que especificó en una regla de transición es 3, la fecha de transición del objeto se calculará como 19/01/2014, a las 00:00 h, UTC.

> **Note**
>
> Amazon S3 solo conserva la última fecha de modificación para cada objeto. Por ejemplo, la consola de Amazon S3 muestra la fecha de Last Modified (Última modificación) en el panel Properties (Propiedades) del objeto. Al crear inicialmente un nuevo objeto, esta fecha refleja el momento de creación del objeto. Si se sustituye el objeto, la fecha cambia según sea necesario. Por lo tanto, la fecha de creación es sinónimo de la fecha de Última modificación.

Al especificar el número de días en las acciones `NoncurrentVersionTransition` y `NoncurrentVersionExpiration` de una configuración de ciclo de vida, tenga en cuenta lo siguiente:

- El valor que especifica es el número de días a partir del momento en el que la versión del objeto deja de ser actual (es decir, desde que se sobrescribe o se elimina un objeto), tras los cuales Amazon S3 realizará la acción en el objeto o los objetos especificados.
- Amazon S3 calcula este momento agregando el número de días especificados en la regla a la hora en la que se creó la nueva versión sucesora del objeto, y redondea la hora resultante a la medianoche del día siguiente, hora UTC. Por ejemplo, supongamos que en su bucket tiene la versión actual de un objeto creado el 01/01/2014, a las 10:30 a. m. UTC. Si la nueva versión

del objeto que sustituye a la versión actual se creó el 15/01/2014 a las 10:30 a. m. UTC y se especifican tres días en una regla de transición, la fecha de transición del objeto calculada será 19/01/2014, a las 00:00 UTC.

Reglas de ciclo de vida: basadas en una fecha específica

Al especificar una acción en una regla del ciclo de vida de S3, puede especificar la fecha en la que desea que Amazon S3 realice la acción. Cuando llegue la fecha específica, Amazon S3 aplicará la acción a todos los objetos cualificados (según los criterios de filtrado).

Si especifica una acción de S3 Lifecycle con una fecha en el pasado, todos los objetos cualificados serán inmediatamente aptos para esa acción de Lifecycle.

> **Important**
>
> La acción basada en la fecha no es una acción puntual. Amazon S3 seguirá aplicando la acción basada en la fecha incluso después de que esta haya pasado, siempre que el estado de la regla sea `Enabled`.
> Por ejemplo, supongamos que especifica una acción de `Expiration` basada en una fecha para eliminar todos los objetos (y supongamos también que la regla no especifica ningún filtro). En la fecha especificada, Amazon S3 hace que venzan todos los objetos del bucket. Amazon S3 también seguirá haciendo que venzan los nuevos objetos creados en el bucket. Para detener la acción de Lifecycle, debe eliminar la acción de la regla de Lifecycle, deshabilitar la regla o eliminar la regla de la configuración de Lifecycle.

El valor de la fecha debe estar en formato ISO 8601. La hora siempre es medianoche UTC.

> **Note**
>
> No puede crear reglas de Lifecycle basadas en una fecha con la consola de Amazon S3, pero sí que puede usarla para ver, deshabilitar o eliminar esas reglas.

Ejemplos de configuración de S3 Lifecycle

En esta sección se proporcionan ejemplos de configuración de S3 Lifecycle. Cada ejemplo muestra cómo puede especificar el archivo XML en cada uno de los ejemplos de casos.

Important

Cuando tiene varias reglas en una configuración de S3 Lifecycle, un objeto puede reunir los requisitos para varias acciones de S3 Lifecycle realizadas el mismo día. En tales casos, Amazon S3 sigue estas reglas generales:

- La eliminación permanente prevalece sobre la transición.
- La transición prevalece sobre la creación de marcadores de eliminación.
- Cuando un objeto es elegible para una transición S3 Glacier Flexible Retrieval y S3 Standard-IA (o S3 One Zone-IA), Amazon S3 elige la transición S3 Glacier Flexible Retrieval.

Para ver ejemplos, consulte Ejemplo 5: superposición de filtros, conflictos entre acciones de ciclo de vida y lo que hace Amazon S3 con buckets sin control de versiones.

Temas

Ejemplo 1: especificar un filtro

Cada regla de S3 Lifecycle incluye un filtro que puede utilizar para identificar un subconjunto de objetos en el bucket al que se aplica la regla de S3 Lifecycle. Las siguientes configuraciones de ciclo de vida de S3 muestran ejemplos de cómo puede especificar un filtro.

- En esta regla de configuración de S3 Lifecycle, el filtro especifica un prefijo de nombre de clave (`tax/`). Por lo tanto, la regla se aplica a objetos con el prefijo `tax/`, como `tax/doc1.txt` y `tax/doc2.txt`.

  La regla especifica dos acciones que indican a Amazon S3 realizar lo siguiente:

  - Pasar objetos a la clase de almacenamiento S3 Glacier Flexible Retrieval 365 días (un año) después de su creación.
  - Eliminar objetos (la acción `Expiration`) 3650 días (10 años) después de su creación.

```
<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>
```

  En lugar de especificar la antigüedad del objeto en términos de días después de su creación, puede especificar una fecha para cada acción. Sin embargo, no puede usar `Date` y `Days` en la misma regla.

- Si desea que la regla de S3 Lifecycle se aplique a todos los objetos del bucket, especifique un prefijo vacío. En la siguiente configuración, la regla especifica una acción `Transition` que le indica a Amazon S3 que pase los objetos a la clase de almacenamiento S3 Glacier Flexible Retrieval 0 días después de su creación. Esta regla significa que los objetos pueden archivarse en S3 Glacier Flexible Retrieval a la medianoche (UTC) después de su creación. Para obtener más información acerca de las restricciones del ciclo de vida, consulte Restricciones.

```
<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>
```

- Puede especificar cero o un prefijo y cero o más etiquetas de objetos en un filtro. El siguiente ejemplo de código aplica la regla de S3 Lifecycle a un subconjunto de objetos con el prefijo `tax/` y a objetos que tienen dos etiquetas con una clave y valor específicos. Cuando especifica más de un filtro, debe incluir el elemento <And> como se muestra (Amazon S3 aplica un AND lógico para combinar las condiciones del filtro especificadas).

```
...
<Filter>
   <And>
      <Prefix>tax/</Prefix>
      <Tag>
         <Key>key1</Key>
         <Value>value1</Value>
      </Tag>
      <Tag>
         <Key>key2</Key>
         <Value>value2</Value>
      </Tag>
    </And>
</Filter>
...
```

> **Note**
>
> Si tiene uno o más prefijos que comienzan con los mismos caracteres, puede incluir todos esos prefijos en la regla especificando un prefijo parcial sin barra al final (/) en el filtro. Por ejemplo, supongamos que dispone de los siguientes prefijos:
>
> ```
> sales1999/
> sales2000/
> ```

```
sales2001/
```

Para incluir los tres prefijos en la regla, especifique `<Prefix>sales</Prefix>` en la regla de ciclo de vida.

- En lugar de utilizar un prefijo, puede filtrar objetos solo por las etiquetas. Por ejemplo, la siguiente regla de S3 Lifecycle se aplica a objetos que tienen dos etiquetas especificadas (no especifica ningún prefijo).

```
...
<Filter>
   <And>
      <Tag>
         <Key>key1</Key>
         <Value>value1</Value>
      </Tag>
      <Tag>
         <Key>key2</Key>
         <Value>value2</Value>
      </Tag>
    </And>
</Filter>
...
```

Si desea excluir un prefijo concreto de la regla de ciclo de vida, utilice etiquetas para etiquetar todos los objetos de los prefijos que desee incluir en la regla.

Ejemplo 2: desactivación de una regla del ciclo de vida

Puede desactivar una regla de S3 Lifecycle de manera temporal. La siguiente configuración de S3 Lifecycle especifica dos reglas:

- La regla 1 indica a Amazon S3 que realice la transición de los objetos con el prefijo `logs/` a la clase de almacenamiento S3 Glacier Flexible Retrieval inmediatamente después de su creación.
- La regla 2 indica a Amazon S3 que realice la transición de los objetos con el prefijo `documents/` a la clase de almacenamiento S3 Glacier Flexible Retrieval inmediatamente después de su creación.

En la configuración, la regla 1 está habilitada y la regla 2 está deshabilitada. Amazon S3 ignora las reglas desactivadas.

```
<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>
```

Ejemplo 3: Transición entre las clases de almacenamiento durante la vida útil de un objeto

En este ejemplo, se usa la configuración de S3 Lifecycle para pasar objetos a otras clases de almacenamiento durante su vida útil. La transición entre clases de almacenamiento puede ayudar a reducir los costos de almacenamiento. Para obtener más información acerca de los precios, consulte Precios de Amazon S3.

La siguiente configuración de S3 Lifecycle especifica una regla que se aplica a objetos con el prefijo de nombre de clave `logs/`. La regla especifica las siguientes acciones:

- Dos acciones de transición:
  - Pasar objetos a la clase de almacenamiento S3 Standard-IA 30 días después de su creación
  - Pasar objetos a la clase de almacenamiento S3 Glacier Flexible Retrieval 90 días después de su creación.

- Una acción de vencimiento que le indica a Amazon S3 que elimine los objetos un año después de su creación

```
<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

Puede usar una regla para describir todas las acciones de S3 Lifecycle si todas las acciones se aplican al mismo conjunto de objetos (identificado por un filtro). Por otro lado, puede añadir varias reglas que especifiquen de manera individual un filtro diferente.

Important

Cuando tiene varias reglas en una configuración de S3 Lifecycle, un objeto puede reunir los requisitos para varias acciones de S3 Lifecycle realizadas el mismo día. En tales casos, Amazon S3 sigue estas reglas generales:

- La eliminación permanente prevalece sobre la transición.
- La transición prevalece sobre la creación de marcadores de eliminación.

- Cuando un objeto es elegible para una transición S3 Glacier Flexible Retrieval y S3 Standard-IA (o S3 One Zone-IA), Amazon S3 elige la transición S3 Glacier Flexible Retrieval.

Para ver ejemplos, consulte Ejemplo 5: superposición de filtros, conflictos entre acciones de ciclo de vida y lo que hace Amazon S3 con buckets sin control de versiones.

Ejemplo 4: especificar varias reglas

Puede especificar varias reglas si desea que se realicen diferentes acciones de S3 Lifecycle en diferentes objetos. La siguiente configuración de S3 Lifecycle tiene dos reglas:

- La regla 1 se aplica a objetos con el prefijo de nombre de clave `classA/`. Le indica a Amazon S3 que pase los objetos a la clase de almacenamiento S3 Glacier Flexible Retrieval un año después de su creación y que provoque el vencimiento de estos objetos 10 años después de su creación.
- La regla 2 se aplica a objetos con el prefijo de nombre de clave `classB/`. Le indica a Amazon S3 que pase los objetos a la clase de almacenamiento S3 Standard-IA 90 días después de su creación y que los elimine un año después de su creación.

```
<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

Cuando tiene varias reglas en una configuración de S3 Lifecycle, un objeto puede reunir los requisitos para varias acciones de S3 Lifecycle realizadas el mismo día. En tales casos, Amazon S3 sigue estas reglas generales:

- La eliminación permanente prevalece sobre la transición.
- La transición prevalece sobre la creación de marcadores de eliminación.
- Cuando un objeto es elegible para una transición S3 Glacier Flexible Retrieval y S3 Standard-IA (o S3 One Zone-IA), Amazon S3 elige la transición S3 Glacier Flexible Retrieval.

Para ver ejemplos, consulte Ejemplo 5: superposición de filtros, conflictos entre acciones de ciclo de vida y lo que hace Amazon S3 con buckets sin control de versiones.

Ejemplo 5: superposición de filtros, conflictos entre acciones de ciclo de vida y lo que hace Amazon S3 con buckets sin control de versiones

Es posible que especifique una configuración de S3 Lifecycle en la que especifique prefijos o acciones superpuestos.

En general, S3 Lifecycle optimiza el costo. Por ejemplo, si dos políticas de vencimiento se superponen, se respeta la política con una fecha de vencimiento anterior para que los datos no permanezcan almacenados más tiempo de lo previsto. Del mismo modo, si dos políticas de

transición se superponen, S3 Lifecycle pasa los objetos a la clase de almacenamiento con costos inferiores.

En ambos casos, S3 Lifecycle intenta elegir la opción que le resulte menos costosa. Una excepción a esta regla general es la clase de almacenamiento S3 Intelligent-Tiering (Capas inteligentes de S3). S3 Intelligent-Tiering se ve favorecido por S3 Lifecycle sobre cualquier clase de almacenamiento, aparte de las clases de almacenamiento de S3 Glacier Flexible Retrieval y S3 Glacier Deep Archive.

En los siguientes ejemplos, se muestra cómo Amazon S3 resuelve posibles conflictos.

Example 1: Superponer prefijos (sin conflicto)

El siguiente ejemplo de configuración tiene dos reglas que especifican prefijos superpuestos de la siguiente manera:

- La primera regla especifica un filtro vacío que denota todos los objetos en el bucket.
- La segunda regla especifica un prefijo de nombre de clave (`logs/`) que solo denota un subconjunto de objetos.

La regla 1 solicita a Amazon S3 que elimine todos los objetos un año después de su creación. La regla 2 solicita a Amazon S3 que realice la transición de un subconjunto de objetos a la clase de almacenamiento S3 Standard-IA 30 días después de su creación.

```
<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>
```

Dado que no hay conflicto en este caso, Amazon S3 hará la transición de los objetos con el prefijo `logs/` a la clase de almacenamiento S3 Standard-IA 30 días después de su creación. Cuando un objeto llegue a un año después de su creación, se eliminará.

Example 2: conflictos entre acciones del ciclo de vida

En este ejemplo de configuración, existen dos reglas que le indican a Amazon S3 que realice dos acciones diferentes en el mismo conjunto de objetos al mismo tiempo durante la vida útil de los objetos:

- Ambas reglas especifican el mismo prefijo de nombre de clave, por lo que ambas reglas se aplican al mismo conjunto de objetos.
- Ambas reglas especifican el mismo momento de aplicación de las reglas, es decir, 365 días después de la creación de los objetos.
- Una regla le indica a Amazon S3 que pase objetos a la clase de almacenamiento S3 Standard-IA y la otra regla desea que Amazon S3 provoque el vencimiento de los objetos al mismo tiempo.

```
<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>
```

En este caso, como usted desea que los objetos venzan (se eliminen), no tiene sentido cambiar la clase de almacenamiento, por lo que Amazon S3 opta por realizar la acción de vencimiento en estos objetos.

Example 3: Superponer prefijos causa conflictos entre acciones del ciclo de vida

En este ejemplo, la configuración tiene dos reglas que especifican prefijos superpuestos de la siguiente manera:

- La regla 1 especifica un prefijo vacío (que denota todos los objetos).
- La regla 2 especifica un prefijo de nombre de clave (`logs/`) que identifica un subconjunto de todos los objetos.

Para el subconjunto de objetos con el prefijo de nombre de clave `logs/`, se aplican las acciones de S3 Lifecycle de ambas reglas. Una regla le indica a Amazon S3 que pase los objetos 10 días después de su creación y otra regla le indica a Amazon S3 que pase los objetos 365 días después de su creación.

```
<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>
```

En este caso, Amazon S3 opta por pasarlos 10 días después de su creación.

Example 4: filtros basados en etiquetas y conflictos entre acciones de ciclo de vida resultantes

Suponga que tiene la siguiente configuración de Ciclo de vida de S3 con dos reglas, cada una de las cuales especifica un filtro de etiquetas:

- La regla 1 especifica un filtro basado en etiquetas (`tag1/value1`). Esta regla le indica a Amazon S3 que pase los objetos a la clase de almacenamiento S3 Glacier Flexible Retrieval 365 días después de su creación.
- La regla 2 especifica un filtro basado en etiquetas (`tag2/value2`). Esta regla le indica a Amazon S3 que provoque el vencimiento de los objetos 14 días después de su creación.

La configuración de S3 Lifecycle se muestra en el siguiente ejemplo.

```
<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 objeto tiene ambas etiquetas, Amazon S3 tiene que decidir qué regla seguir. En este caso, Amazon S3 provoca el vencimiento del objeto 14 días después de su creación. El objeto se elimina y, por lo tanto, la acción de transición no se aplica.

> **Important**
>
> Cuando tiene varias reglas en una configuración de S3 Lifecycle, un objeto puede reunir los requisitos para varias acciones de S3 Lifecycle realizadas el mismo día. En tales casos, Amazon S3 sigue estas reglas generales:
>
> - La eliminación permanente prevalece sobre la transición.
> - La transición prevalece sobre la creación de marcadores de eliminación.
> - Cuando un objeto es elegible para una transición S3 Glacier Flexible Retrieval y S3 Standard-IA (o S3 One Zone-IA), Amazon S3 elige la transición S3 Glacier Flexible Retrieval.
>
> Para ver ejemplos, consulte Ejemplo 5: superposición de filtros, conflictos entre acciones de ciclo de vida y lo que hace Amazon S3 con buckets sin control de versiones.

Ejemplo 6: especificar una regla del ciclo de vida para un bucket habilitado para el control de versiones

Suponga que tiene un bucket habilitado para el control de versiones, lo cual significa que para cada objeto, tiene una versión actual y cero o más versiones no actuales. (Para obtener más

información sobre S3 Versioning, consulte Usar el control de versiones en buckets de S3). En este ejemplo, desea mantener el valioso historial de un año y eliminar las versiones no actuales. Las configuraciones de S3 Lifecycle admiten mantener de 1 a 100 versiones de cualquier objeto.

Para reducir los costos de almacenamiento, desea trasladar las versiones no actuales a la clase de almacenamiento S3 Glacier Flexible Retrieval 30 días después de que adquieren la condición de ser no actuales (sobre la base de la suposición de estos objetos no actuales son datos inactivos a los que no necesita obtener acceso en tiempo real). Asimismo, espera que disminuya la frecuencia de acceso de las versiones actuales a los 90 días después de su creación, por lo que puede elegir pasar estos objetos a la clase de almacenamiento 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>
```

Ejemplo 7: eliminar marcadores de eliminación de objetos que vencieron

Un bucket habilitado para el control de versiones tiene una versión actual y cero o más versiones no actuales para cada objeto. Cuando elimina un objeto, tenga en cuenta lo siguiente:

- Si no especifica una ID de versión en la solicitud de eliminación, Amazon S3 añade un marcador de eliminación en lugar de eliminar el objeto. La versión actual del objeto se convierte en no actual y el marcador de eliminación se convierte en la versión actual.

- Si especifica una ID de versión en la solicitud de eliminación, Amazon S3 elimina la versión del objeto de manera permanente (no se crea un marcador de eliminación).
- Al marcador de eliminación con cero versiones no actuales se lo denomina marcador de eliminación de objetos que vencieron.

En este ejemplo se muestra un caso en el que se pueden crear marcadores de eliminación de objetos que vencieron en el bucket y cómo puede usar la configuración de S3 Lifecycle para indicarle a Amazon S3 que elimine los marcadores de eliminación de objetos que vencieron.

Suponga que escribe una configuración de S3 Lifecycle que utiliza la acción `NoncurrentVersionExpiration` para eliminar las versiones no actuales 30 días después de que se conviertan en no actuales y retiene como máximo 10 versiones no actuales, tal como se muestra en el ejemplo que sigue.

```
<LifecycleConfiguration>
    <Rule>
        ...
        <NoncurrentVersionExpiration>
            <NewerNoncurrentVersions>10</NewerNoncurrentVersions>
            <NoncurrentDays>30</NoncurrentDays>
        </NoncurrentVersionExpiration>
    </Rule>
</LifecycleConfiguration>
```

La acción `NoncurrentVersionExpiration` no se aplica a las versiones actuales del objeto. Elimina solo las versiones no actuales.

En el caso de las versiones actuales de los objetos, tiene las siguientes opciones para administrar su vida útil según si las versiones actuales de los objetos siguen un ciclo de vida bien definido:

- Las versiones actuales de los objetos siguen un ciclo de vida bien definido.

  En este caso, puede utilizar una política de Ciclo de vida de S3 con la acción `Expiration` para indicarle a Amazon S3 que elimine las versiones actuales, tal como se muestra en el siguiente ejemplo.

  ```
  <LifecycleConfiguration>
      <Rule>
          ...
  ```

```
        <Expiration>
           <Days>60</Days>
        </Expiration>
        <NoncurrentVersionExpiration>
            <NewerNoncurrentVersions>10</NewerNoncurrentVersions>
            <NoncurrentDays>30</NoncurrentDays>
        </NoncurrentVersionExpiration>
    </Rule>
</LifecycleConfiguration>
```

En este ejemplo, Amazon S3 elimina las versiones actuales 60 días después de que se crearon agregando un marcador de eliminación para cada versión actual del objeto. Este proceso convierte la versión actual en no actual, y el marcador de eliminación se convierte en la versión actual. Para obtener más información, consulte Usar el control de versiones en buckets de S3.

> **Note**
>
> No puede especificar una etiqueta `Days` y `ExpiredObjectDeleteMarker` en la misma regla. Al especificar la etiqueta `Days`, Amazon S3 limpiará automáticamente `ExpiredObjectDeleteMarker` cuando los marcadores de eliminación sean lo suficientemente antiguos como para cumplir con los criterios de antigüedad. Para limpiar los marcadores de eliminación tan pronto como se conviertan en la única versión, cree una regla independiente con solo la etiqueta `ExpiredObjectDeleteMarker`.

La acción `NoncurrentVersionExpiration` en la misma configuración de S3 Lifecycle elimina los objetos no actuales 30 días después de que adquieren la condición de ser no actuales.
Por lo tanto, en este ejemplo todas las versiones de objetos se eliminan permanentemente 90 días después de la creación del objeto. Si bien durante este proceso se crean marcadores de eliminación de objetos vencidos, Amazon S3 detecta y elimina los marcadores de eliminación de objetos vencidos por usted.

- Las versiones actuales de los objetos no tienen un ciclo de vida bien definido.

  En este caso, puede eliminar los objetos manualmente cuando no los necesita y así crear un marcador de eliminación con una o más versiones no actuales. Si la configuración de S3 Lifecycle con la acción `NoncurrentVersionExpiration` elimina todas las versiones no actuales, ahora tiene marcadores de eliminación de objetos que vencieron.

Específicamente para este caso, la configuración de S3 Lifecycle proporciona una acción `Expiration` que puede utilizar para eliminar los marcadores de eliminación de objetos que vencieron.

```
<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>
```

Al establecer el elemento `ExpiredObjectDeleteMarker` como `true` en la acción `Expiration`, le indica a Amazon S3 que elimine los marcadores de eliminación de objetos que vencieron.

> **Note**
>
> Cuando utiliza la acción de S3 Lifecycle de `ExpiredObjectDeleteMarker`, la regla no puede especificar un filtro basado en etiquetas.

Ejemplo 8: configuración del ciclo de vida para anular cargas multipartes

Puede utilizar las operaciones de la API REST de carga multiparte de Amazon S3 para cargar objetos grandes en partes. Para obtener más información acerca de las cargas multipartes, consulte Carga y copia de objetos con la carga multiparte.

Mediante la configuración de S3 Lifecycle, puede indicarle a Amazon S3 que detenga las cargas multipartes incompletas (identificadas por el prefijo de nombre de clave especificado en la regla) si

no se completan en una cantidad de días especificada después de iniciarse. Cuando Amazon S3 anula una carga multiparte, elimina todas las partes asociadas con la carga multiparte. Este proceso ayuda a controlar los costos de almacenamiento asegurándose de que no tenga cargas multipartes incompletas con partes que se almacenan en Amazon S3.

> **Note**
>
> Cuando utiliza la acción de S3 Lifecycle de `AbortIncompleteMultipartUpload`, la regla no puede especificar un filtro basado en etiquetas.

El siguiente es un ejemplo de configuración de S3 Lifecycle que especifica una regla con la acción `AbortIncompleteMultipartUpload`. Esta acción le indica a Amazon S3 que detenga las cargas multipartes incompletas siete días después de iniciarse.

```
<LifecycleConfiguration>
    <Rule>
        <ID>sample-rule</ID>
        <Filter>
           <Prefix>SomeKeyPrefix/</Prefix>
        </Filter>
        <Status>rule-status</Status>
        <AbortIncompleteMultipartUpload>
          <DaysAfterInitiation>7</DaysAfterInitiation>
        </AbortIncompleteMultipartUpload>
    </Rule>
</LifecycleConfiguration>
```

Ejemplo 9: Configuración de Lifecycle mediante reglas basadas en tamaño

Puede crear reglas que realicen una transición de objetos basándose únicamente en su tamaño. Puede especificar un tamaño mínimo (`ObjectSizeGreaterThan`) o un tamaño máximo (`ObjectSizeLessThan`), o bien puede especificar un rango de tamaños de objeto en bytes. Cuando utilice más de un filtro como un prefijo y una regla de tamaño, debe envolver los filtros en un elemento `<And>`.

> **Note**
>
> Los filtros `ObjectSizeGreaterThan` y `ObjectSizeLessThan` excluyen los valores especificados. Para obtener más información, consulte the section called “Elemento Filter”.

```
<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>
```

Cuando especifica un rango utilizando los elementos `ObjectSizeGreaterThan` y `ObjectSizeLessThan`, el tamaño máximo de objeto debe ser superior al tamaño mínimo de objeto. Cuando utilice más de un filtro, debe envolver los filtros en un elemento <And>. En el siguiente ejemplo, se muestra cómo especificar objetos en un rango de entre 500 y 64 000 bytes.

```
<LifecycleConfiguration>
    <Rule>
        ...
          <And>
             <ObjectSizeGreaterThan>500</ObjectSizeGreaterThan>
             <ObjectSizeLessThan>64000</ObjectSizeLessThan>
          </And>
    </Rule>
</LifecycleConfiguration>
```

También puede crear reglas para caducar específicamente objetos no actuales que no tengan datos, incluidos los objetos de marcador de eliminación no actuales creados en un bucket con control de

versiones habilitado. En el siguiente ejemplo se utiliza la acción `NoncurrentVersionExpiration` para eliminar las versiones no actuales 30 días después de que se conviertan en no actuales y retiene como máximo 10 versiones no actuales de objetos. También utiliza el elemento `ObjectSizeLessThan` para filtrar únicamente objetos sin datos.

```
<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>
```

Inventario de Amazon S3

> **Important**
>
> Amazon S3 aplica ahora el cifrado del servidor con claves administradas por Amazon S3 (SSE-S3) como el nivel básico de cifrado para cada bucket de Amazon S3. Desde el 5 de enero de 2023, todas las cargas de objetos nuevos a Amazon S3 se cifran automáticamente sin costo adicional y sin afectar al rendimiento. El estado de cifrado automático para la configuración de cifrado predeterminada en el bucket de S3 y para cargas de objetos nuevos está disponible en registros de AWS CloudTrail, Inventario de S3, Lente de almacenamiento de S3, la consola de Amazon S3 y como encabezado de respuesta a la API de Amazon S3 adicional en AWS Command Line Interface y los SDK de AWS. Para obtener más información, consulte Preguntas frecuentes del cifrado predeterminado.

Puede usar el Inventario de Amazon S3 para administrar su almacenamiento. Por ejemplo, puede utilizarlo para auditar e informar sobre el estado de replicación y cifrado de los objetos para sus necesidades empresariales, de conformidad y legales. Puede simplificar o acelerar los flujos de trabajo empresariales y los trabajos relacionados con los macrodatos mediante el Inventario de

Amazon S3, que ofrece una alternativa programada a las operaciones de la API `List` sincrónica de Amazon S3. El Inventario de Amazon S3 no utiliza operaciones de API `List` para auditar los objetos y no afecta a la tasa de solicitudes del bucket.

El Inventario de Amazon S3 proporciona archivos de salida con formato de valores separados por comas (CSV), Optimized Row Columnar (ORC) de Apache o Apache Parquet que muestran diaria o semanalmente los objetos y los metadatos correspondientes en un bucket de S3 u objetos con un prefijo compartido (es decir, objetos con nombres que comienzan con la misma cadena). Si configura un inventario semanal, se genera un informe cada domingo (zona horaria UTC) después del informe inicial. Para obtener más información acerca de los precios de Amazon S3 Inventory, consulte Precios de Amazon S3.

Puede configurar varias listas de inventario para un bucket. Al configurar una lista de inventario, puede especificar lo siguiente:

- Qué metadatos de objetos incluir en el inventario
- Si se deben enumerar todas las versiones del objeto o solo las versiones actuales
- Dónde almacenar la salida del archivo de lista de inventario
- Si se debe generar el inventario de forma diaria o semanal
- Si se debe cifrar el archivo de lista de inventario

Puede consultar el Inventario de Amazon S3 con consultas SQL estándar mediante Amazon Athena, Amazon Redshift Spectrum y otras herramientas, como Presto, Apache Hive y Apache Spark. Para obtener más información acerca del uso de Athena para consultar sus archivos de inventario, consulte the section called “Consultas de inventario con Athena”.

Buckets de origen y destino

El bucket para el que el inventario enumera los objetos se denomina bucket de origen. El bucket en el que se almacena el archivo con la lista del inventario se denomina bucket de destino.

Bucket de origen

El inventario enumera los objetos almacenados en el bucket de origen. Puede obtener una lista de inventario para todo un bucket, o puede filtrar la lista por prefijo de nombre de clave de objeto.

El bucket de origen:

- Contiene los objetos enumerados en el inventario.

- Contiene la configuración del inventario.

Bucket de destino

Los archivos con la lista de Amazon S3 Inventory se escriben en el bucket de destino. Para agrupar todos los archivos de la lista de inventario en una ubicación común del bucket de destino, puede especificar un prefijo de destino en la configuración del inventario.

El bucket de destino:

- Contiene las listas de archivos de inventario.
- Contiene los archivos de manifiesto que enumeran todos los archivos de lista de inventario almacenados en el bucket de destino. Para obtener más información, consulte Manifiesto de inventario.
- Debe tener una política de bucket para conceder a Amazon S3 permiso para verificar la propiedad del bucket y permiso para escribir archivos en el bucket.
- Debe estar en la misma Región de AWS que el bucket de origen.
- Puede ser igual que la del bucket de origen.
- Puede ser propiedad de una Cuenta de AWS diferente a la cuenta que es propietaria del bucket de origen.

Lista de Amazon S3 Inventory

Un archivo de lista de inventario contiene una lista de los objetos del bucket de origen y los metadatos para cada objeto. En el bucket de destino, se almacena un archivo de lista de inventario con uno de los siguientes formatos:

- Un archivo CSV comprimido con GZIP
- Como archivo Optimized Row Columnar (ORC) de Apache comprimido con ZLIB
- Como archivo de Apache Parquet comprimido con Snappy

> **Note**
>
> No se garantiza que los objetos de los informes de inventario de Amazon S3 estén clasificados en ningún orden.

Un archivo de lista de inventario contiene una lista de los objetos del bucket de origen y los metadatos para cada objeto de esa lista.

- Nombre del bucket: el nombre del bucket para el que se realiza el inventario.
- Nombre de clave: el nombre de la clave de objeto (o clave) que identifica unívocamente el objeto del bucket. Cuando se utiliza el formato de archivo CSV, el nombre de la clave se codifica como URL y se debe descodificar antes de poder utilizarlo.
- ID de versión: el ID de la versión del objeto. Si activa el control de versiones en un bucket, Amazon S3 asigna un número de versión a los objetos que agregue a dicho bucket. Para obtener más información, consulte Usar el control de versiones en buckets de S3. (Este campo no estará incluido si la lista solo está configurada para la versión actual de los objetos).
- IsLatest: estará establecido en `True` si el objeto es la versión actual del objeto. (Este campo no estará incluido si la lista solo está configurada para la versión actual de los objetos).
- Marcador de eliminación: establecido en `True`, si el objeto es un marcador de eliminación. Para obtener más información, consulte Usar el control de versiones en buckets de S3. (Este campo se añade automáticamente al informe si este se ha configurado para que incluya todas las versiones de los objetos).
- Tamaño: el tamaño del objeto en bytes, sin incluir el tamaño de las cargas multipartes incompletas, los metadatos del objeto ni los marcadores de eliminación.
- Fecha de la última modificación: la fecha de creación del objeto o la última fecha de modificación, la última existente.
- ETag: la etiqueta de entidad (ETag) es un hash del objeto. La ETag solo refleja los cambios en el contenido de un objeto, no en sus metadatos. La ETag puede ser un resumen MD5 de los datos del objeto. Esto dependerá del método de creación del objeto y del tipo de cifrado.
- Clase de almacenamiento: la clase de almacenamiento utilizada para almacenar el objeto. Establecida en `STANDARD`, `REDUCED_REDUNDANCY`, `STANDARD_IA`, `ONEZONE_IA`, `INTELLIGENT_TIERING`, `GLACIER`, `DEEP_ARCHIVE`, `OUTPOSTS`, `GLACIER_IR` o `SNOW`. Para obtener más información, consulte Uso de las clases de almacenamiento de Amazon S3.
- Marcador de carga multiparte: establecido en `True` si el objeto se cargó mediante una carga multiparte. Para obtener más información, consulte Carga y copia de objetos con la carga multiparte.
- Estado de reproducción: establecido en `PENDING`, `COMPLETED`, `FAILED` o `REPLICA`. Para obtener más información, consulte Obtención de información del estado de replicación.
- Estado de cifrado: el estado del cifrado del servidor dependiendo del tipo de clave de cifrado que se utilice: cifrado del servidor con claves administradas por Amazon S3 (SSE-S3), cifrado

del servidor con claves de AWS Key Management Service (AWS KMS) (SSE-KMS), cifrado del servidor de doble capa con claves de AWS KMS (DSSE-KMS) o cifrado del servidor con claves proporcionadas por el cliente (SSE-C). Establecida en `SSE-S3`, `SSE-KMS`, `DSSE-KMS`, `SSE-C` o `NOT-SSE`. Un estado de `NOT-SSE` significa que el objeto no está cifrado con el cifrado del servidor. Para obtener más información, consulte Protección de los datos mediante el cifrado.

- Fecha de retención del bloqueo de objetos de S3: la fecha hasta la que no se puede eliminar un objeto bloqueado. Para obtener más información, consulte Usar Bloqueo de objetos de S3.
- Modo de retención del bloqueo de objetos de S3: establecido en `Governance` o `Compliance` para los objetos que están bloqueados. Para obtener más información, consulte Usar Bloqueo de objetos de S3.
- Estado de retención legal del bloqueo de objetos de S3: establecido en `On` si se ha aplicado una retención legal a un objeto. De lo contrario, se establece en `Off`. Para obtener más información, consulte Usar Bloqueo de objetos de S3.
- Capa de acceso de S3 Intelligent-Tiering: capa de acceso (frecuente o infrecuente) del objeto si está almacenado en la clase de almacenamiento S3 Intelligent-Tiering. Establecida en `FREQUENT`, `INFREQUENT`, `ARCHIVE_INSTANT_ACCESS`, `ARCHIVE` o `DEEP_ARCHIVE`. Para obtener más información, consulte Clase de almacenamiento para optimizar automáticamente los datos con patrones de acceso cambiantes o desconocidos.
- Estado de clave de bucket de S3: establecido en `ENABLED` o `DISABLED`. Indica si el objeto utiliza una clave de bucket de S3 para SSE-KMS. Para obtener más información, consulte Uso de claves de bucket de Amazon S3.
- Algoritmo de suma de comprobación: indica el algoritmo utilizado para crear la suma de comprobación del objeto.
- Lista de control de acceso a objetos: una lista de control de acceso (ACL) para cada objeto que define a qué Cuentas de AWS o grupos se les concede acceso a este objeto y el tipo de acceso que se concede. El campo ACL del objeto se define en formato JSON. Un informe de inventario de S3 incluye las ACL asociadas a los objetos del bucket de origen, incluso cuando las ACL están deshabilitadas para el bucket. Para obtener más información, consulte Uso del campo de objeto de ACL y Información general de las Listas de control de acceso (ACL).

> **Note**
>
> El campo ACL del objeto se define en formato JSON. Un informe de inventario muestra el valor del campo ACL del objeto como una cadena codificada en base64.
> Por ejemplo, supongamos que tiene el siguiente campo ACL de objeto en formato JSON:

```
{
        "version": "2022-11-10",
        "status": "AVAILABLE",
        "grants": [{
            "canonicalId": "example-canonical-user-ID",
            "type": "CanonicalUser",
            "permission": "READ"
        }]
}
```

El campo ACL del objeto está codificado y se muestra como la siguiente cadena codificada en base64:

```
eyJ2ZXJzaW9uIjoiMjAyMi0xMS0xMCIsInN0YXR1cyI6IkFWQUlMQUJMRSIsImdyYW50cyI6W3siY2Fub25pY2Fs
```

Para obtener el valor descodificado en JSON para el campo ACL del objeto, puede consultar este campo en Amazon Athena. Para obtener consultas de ejemplo, consulte Consulta de Amazon S3 Inventory con Amazon Athena.

- Propietario del objeto: el propietario del objeto.

> **Note**
>
> Cuando un objeto llega al final de su vida útil según su configuración de ciclo de vida, Amazon S3 lo coloca en una cola para eliminarlo de manera asincrónica. Por ello, es posible que haya un desfase entre la fecha de vencimiento y la fecha en que Amazon S3 elimina un objeto. El informe de inventario incluye los objetos que han caducado pero que aún no se han retirado. Para obtener más información sobre las acciones de vencimiento en S3 Lifecycle, consulte Vencimiento de objetos.

Le recomendamos que cree una política del ciclo de vida que elimine las listas de inventario antiguas. Para obtener más información, consulte Administración del ciclo de vida del almacenamiento.

El permiso `s3:PutInventoryConfiguration` permite al usuario seleccionar todos los campos de metadatos que aparecen anteriormente para cada objeto al configurar una lista de inventario y

especificar el bucket de destino para almacenar el inventario. Un usuario con acceso de lectura a los objetos del bucket de destino puede acceder a todos los campos de metadatos de objetos que están disponibles en la lista de inventario. Para restringir el acceso a un informe de inventario, consulte Concesión de permisos para el inventario de S3 y el análisis de S3.

Consistencia del inventario

Puede que en cada lista de inventario no aparezcan todos sus objetos. La lista de inventario ofrece consistencia final para las solicitudes PUT tanto de nuevos objetos como de objetos sobrescritos, además de las solicitudes `DELETE`. Cada lista de inventario de un bucket es una instantánea de los objetos del bucket. En última instancia, estas listas son consistentes (es decir, una lista puede no incluir objetos añadidos o eliminados recientemente).

Para comprobar el estado de un objeto antes de realizar ninguna acción sobre él, le recomendamos que realice una solicitud `HeadObject` con la API de REST para recuperar los metadatos del objeto o para comprobar sus propiedades en la consola de Amazon S3. También puede verificar los metadatos del objeto con la AWS CLI o los SDK de AWS. Para obtener más información, consulte `HeadObject` en la Referencia de la API de Amazon Simple Storage Service.

Para obtener más información sobre cómo trabajar con Amazon S3 Inventory, consulte los siguientes temas.

Temas

- Configuración de Inventario de Amazon S3
- Configuración de notificaciones de eventos de Amazon S3 para completar el inventario
- Localizar la descripción del inventario
- Consulta de Amazon S3 Inventory con Amazon Athena
- Convertir cadenas de ID de versión vacías de informes de Amazon S3 Inventory en cadenas nulas
- Uso del campo de objeto de ACL

Configuración de Inventario de Amazon S3

Inventario de Amazon S3 proporciona una lista de archivos sin formato de los objetos y metadatos en una programación definida por usted. Puede usar Inventario de S3 como una programación alternativa de la operación de la API `List` síncrona de Amazon S3. S3 Inventory proporciona archivos de salida de valores separados por comas (CSV), Apache optimized row columnar (ORC) o Apache Parquet (Parquet) que enumeran sus objetos y sus metadatos correspondientes.

Puede configurar Inventario de S3 para crear listas de inventario diaria o semanalmente para un bucket de S3 o para objetos que comparten un prefijo (objetos con nombres que comienzan con la misma cadena). Para obtener más información, consulte Inventario de Amazon S3.

En esta sección se describe cómo configurar un inventario, incluida información detallada acerca de los buckets de origen y destino del inventario.

Temas

Información general

Amazon S3 Inventory le ayuda a administrar su almacenamiento creando listas de los objetos que hay en un bucket de S3 en un periodo definido. Puede configurar varias listas de inventario para un bucket. Las listas del inventario se publican en archivos CSV, ORC o Parquet en un bucket de destino.

La forma más sencilla de configurar un inventario es a través de la consola de Amazon S3, pero también puede utilizar la API de REST de Amazon S3, la AWS Command Line Interface (AWS CLI) o los SDK de AWS. La consola realiza el primer paso del siguiente procedimiento automáticamente: añadir un política de bucket al bucket de destino.

Para configurar Amazon S3 Inventory para un bucket de S3

1. Agregue una política de bucket para el bucket de destino.

   Debe crear una política de buckets en el bucket de destino que conceda permisos para que Amazon S3 escriba objetos en el bucket en la ubicación definida. Para ver una política de ejemplo, consulte Concesión de permisos para el inventario de S3 y el análisis de S3.
2. Configure un inventario para enumerar los objetos de un bucket de origen y publicar la lista en un bucket de destino.

Al configurar una lista de inventario para un bucket de origen, debe especificar el bucket de destino en el que quiera que se almacene la lista, y si desea generar la lista cada día o cada semana. También puede configurar si desea enumerar todas las versiones de los objetos o solo las actuales y qué metadatos de los objetos incluir.

Algunos campos de metadatos de objetos en las configuraciones de los informes de Inventario de S3 son opcionales, lo que significa que están disponibles de manera predeterminada pero pueden restringirse cuando se concede el permiso `s3:PutInventoryConfiguration` a un usuario. Puede controlar si los usuarios pueden incluir estos campos de metadatos opcionales en sus informes mediante la clave de condición `s3:InventoryAccessibleOptionalFields`.

Para obtener más información acerca de los campos de metadatos opcionales disponibles en Inventario de S3, consulte OptionalFields en la Referencia de la API de Amazon Simple Storage Service. Para obtener más información sobre cómo restringir el acceso a determinados campos de metadatos opcionales en una configuración de inventario, consulte Control de la creación de la configuración del informe de inventario de S3.

Puede especificar que el archivo de lista de inventario se cifre mediante el cifrado del servidor con una clave administrada de Amazon S3 (SSE-S3) o con una clave administrada por el cliente de AWS Key Management Service (AWS KMS).

> **Note**
>
> La Clave administrada de AWS (`aws/s3`) no es compatible para el cifrado SSE-KMS con Inventario de S3.

Para obtener más información sobre SSE-S3 y SSE-KMS, consulte Protección de los datos con el cifrado del servidor. Si va a utilizar el cifrado SSE-KMS, consulte el paso 3.

- Para obtener información acerca de cómo utilizar la consola para configurar una lista de inventario, consulte Configuración del inventario mediante la consola de S3.
- Para utilizar la API de Amazon S3 para configurar una lista de inventario, use la operación de API de REST PutBucketInventoryConfiguration o su equivalente desde la AWS CLI o los SDK de AWS.

3. Para cifrar el archivo de lista de inventario con SSE-KMS, conceda permiso a Simple Storage Service (Amazon S3) para utilizar AWS KMS key.

Puede configurar el cifrado del archivo de la lista de inventario utilizando la consola de Amazon S3, la API de REST de Amazon S3, la AWS CLI o los SDK de AWS. Cualquiera sea el método que elija, debe conceder permiso a Amazon S3 para utilizar la clave administrada por el cliente con el fin de cifrar el archivo de inventario. Para conceder permiso a Amazon S3, se modifica la política de claves para la clave administrada por el cliente que desea utilizar para cifrar el archivo de inventario. Para obtener más información, consulte Concesión de permiso a Amazon S3 con el fin de utilizar su clave administrada por el cliente para el cifrado.

El bucket de destino que almacena el archivo de la lista de inventario puede ser propiedad de otra Cuenta de AWS que la cuenta que posee el bucket de origen. Si utiliza el cifrado SSE-KMS para las operaciones entre cuentas de Inventario de Amazon S3, le recomendamos que utilice un ARN de clave de KMS totalmente cualificado cuando configure el inventario de S3. Para obtener más información, consulte Uso del cifrado SSE-KMS para operaciones entre cuentas y ServerSideEncryptionByDefault en la Referencia de la API de Amazon Simple Storage Service.

Creación de una política de bucket de destino

Si crea la configuración de inventario a través de la consola de Amazon S3, Amazon S3 crea automáticamente una política de bucket en el bucket de destino que concede permisos de escritura de Amazon S3 al bucket. Sin embargo, si crea la configuración de inventario a través de la AWS CLI, los SDK de AWS o la API de REST de Amazon S3, debe añadir manualmente una política de buckets en el bucket de destino. Para obtener más información, consulte Concesión de permisos para el inventario de S3 y el análisis de S3. La política de buckets de destino de Inventario de S3 permite a Amazon S3 escribir en el bucket los datos para los informes de inventario.

Si se produce un error al intentar crear la política de bucket, recibirá instrucciones para solucionarlo. Por ejemplo, si elige un bucket de destino en otra Cuenta de AWS y no tiene permisos para leer ni escribir en la política del bucket, aparecerá un mensaje de error.

En este caso, el propietario del bucket de destino debe añadir la política del bucket en el bucket de destino. Si la política no se añade al bucket de destino, no obtendrá un informe de inventario, ya que Amazon S3 no tiene permiso para escribir en el bucket de destino. Si el bucket de origen es propiedad de una cuenta diferente de la del usuario actual, el ID de cuenta correcto del propietario del bucket de origen debe sustituirse en la política.

Concesión de permiso a Amazon S3 con el fin de utilizar su clave administrada por el cliente para el cifrado

Para conceder permiso a Amazon S3 para utilizar su clave administrada por el cliente AWS Key Management Service (AWS KMS) para el cifrado del servidor, debe utilizar una política de claves. Para actualizar su política de claves de modo que pueda utilizar una clave administrada por el cliente, siga el procedimiento que se indica a continuación.

Concesión de permisos de Amazon S3 para cifrar mediante la clave administrada por el cliente

1. Inicie sesión en la Cuenta de AWS con la AWS Management Console propietaria de la clave administrada por el cliente.
2. Abra la consola de AWS KMS en https://console.aws.amazon.com/kms.
3. Para cambiar la Región de AWS, utilice el Selector de regiones ubicado en la esquina superior derecha de la página.
4. En el panel de navegación izquierdo, elija Customer managed keys (Claves administrada por el cliente).
5. En Claves administradas por el cliente, seleccione la clave administrada por el cliente que desee usar para cifrar los archivos de inventario.
6. En la sección Key policy (Política de claves), elija Switch to policy view (Cambiar a la vista de política).
7. Para actualizar la política de claves, elija Editar.
8. En la página Editar política de claves, añada la siguiente política de claves a la política de claves existente. Para *source-account-id* y *amzn-s3-demo-source-bucket*, proporcione los valores adecuados para su caso de uso.

```
{
    "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:::amzn-s3-demo-source-bucket"
        }
     }
}
```

9. Elija Guardar cambios.

Para obtener más información acerca de la creación de claves de administradas por el cliente y del uso de políticas de claves, consulte los siguientes enlaces en la AWS Key Management Service Guía de desarrolladores:

- Administración de claves
- Políticas de claves en AWS KMS

Configuración del inventario mediante la consola de S3

Siga estas instrucciones para configurar el inventario mediante la consola de S3.

> **Note**
>
> Amazon S3 puede tardar hasta 48 horas en entregar el primer informe del inventario.

1. Inicie sesión AWS Management Console Management Console y abra la consola de Amazon S3 en https://console.aws.amazon.com/s3/.
2. En el panel de navegación izquierdo, elija Instancias. En la lista Buckets, elija el nombre del bucket para el que desea configurar el Inventario de Amazon S3.
3. Seleccione la pestaña Management.
4. En Inventory configurations (Configuraciones de inventario), seleccione Create inventory configuration (Crear configuración de inventario).
5. En Nombre de configuración de inventario, escriba un nombre.
6. En Ámbito del inventario, haga lo siguiente:
   - Escriba un prefijo opcional.

- Elija las versiones de objeto que desea incluir: Solo versiones actuales o Incluir todas las versiones.

7. En Report details (Detalles del informe), elija la ubicación de la Cuenta de AWS en la que desea guardar los informes: This account (Esta cuenta) o A different account (Una cuenta diferente).
8. Seleccione el bucket de destino en el que desea guardar los informes de inventario en Destino.

   El bucket de destino debe estar en la misma Región de AWS que el bucket para el que configura el inventario. El bucket de destino puede estar en una Cuenta de AWS diferente. Al especificar el bucket de destino, también puede incluir un prefijo opcional para agrupar sus informes de inventario.

   En el campo de bucket Destino verá la instrucción Permiso del bucket de destino que se añade a la política del bucket de destino para permitir que Amazon S3 coloque datos en ese bucket. Para obtener más información, consulte Creación de una política de bucket de destino.
9. En Frecuencia, elija la frecuencia con la que se generará el informe: Diario o Semanal.
10. En Formato de salida, elija uno de los siguientes formatos para el informe:

    - CSV: si tiene previsto utilizar este informe de inventario con Operaciones por lotes de S3 o si desea analizar este informe en otra herramienta, como Microsoft Excel, elija CSV.
    - Apache ORC
    - Apache Parquet
11. En Status (Estado), seleccione Enable (Activar) o Disable (Desactivar).
12. Para configurar el cifrado del servidor, siga los siguientes pasos en Cifrado del informe de inventario:

    a. En Cifrado del servidor, elija No especifique una clave de cifrado o Especificar una clave de cifrado para cifrar datos.

       - Para mantener la configuración del bucket para el cifrado predeterminado del servidor de los objetos al almacenarlos en Amazon S3, elija No especifique una clave de cifrado. Siempre y cuando el bucket de destino tenga habilitadas las claves de bucket de S3, la operación de copia aplicará una clave de bucket de S3 al bucket de destino.

         > **Note**
         >
         > Si la política de buckets para el destino especificado exige que los objetos estén cifrados antes de almacenarlos en Amazon S3, debe elegir Especificar una

clave de cifrado. De lo contrario, se producirá un error al copiar los objetos en el destino.

- Para cifrar objetos antes de almacenarlos en Amazon S3, elija Especificar una clave de cifrado.

b. Si elige Especificar una clave de cifrado, en Tipo de cifrado, debe elegir entre Clave administrada por Amazon S3 (SSE-S3) o Clave de AWS Key Management Service (SSE-KMS).

SSE-S3 utiliza uno de los cifrados de bloques más seguros, Advanced Encryption Standard de 256 bits (AES-256), para cifrar cada objeto. SSE-KMS le proporciona más control sobre su clave. Para obtener más información sobre SSE-S3, consulte Uso del cifrado del servidor con claves administradas por Amazon S3 (SSE-S3). Para obtener más información sobre SSE-KMS, consulte Uso del cifrado del servidor con claves de AWS KMS (SSE-KMS).

> **Note**
>
> Para cifrar el archivo de lista de inventario con SSE-KMS, debe conceder permiso a Amazon S3 para utilizar la clave administrada por el cliente. Para ver instrucciones, consulte Conceder permiso a Amazon S3 para cifrar mediante sus claves de KMS.

c. Si elige la Clave de AWS Key Management Service (SSE-KMS), en AWS KMS key, puede especificar su clave AWS KMS mediante una de las siguientes opciones.

> **Note**
>
> Si el bucket de destino que almacena el archivo de la lista de inventario pertenece a otra Cuenta de AWS, asegúrese de que utiliza un ARN de clave de KMS totalmente cualificado para especificar su clave de KMS.

- Para elegir entre una lista de claves de KMS disponibles, seleccione Elija entre sus claves de AWS KMS y elija una clave de KMS de cifrado simétrico en la lista de claves disponibles. Asegúrese de que la clave de KMS esté en la misma región que su bucket.

> **Note**
>
> En esta lista aparecen tanto la Clave administrada de AWS (`aws/s3`) como las claves administradas por el cliente. Sin embargo, la Clave administrada de AWS (`aws/s3`) no es compatible para el cifrado SSE-KMS con Inventario de S3.

- Para introducir el ARN de la clave de KMS, elija Introducir el ARN de la clave de AWS KMS e introduzca el ARN de la clave de KMS en el campo que aparece.
- Para crear una nueva clave administrada por el cliente en la consola de AWS KMS, elija Crear una clave de KMS.

13. En Campos de metadatos adicionales, seleccione uno o más de los siguientes campos opcionales para añadirlos al informe de inventario:

    - Tamaño: el tamaño del objeto en bytes, sin incluir el tamaño de las cargas multipartes incompletas, los metadatos del objeto ni los marcadores de eliminación.
    - Fecha de la última modificación: la fecha de creación del objeto o la última fecha de modificación, la última existente.
    - Multipart upload (Carga multiparte): especifica que el objeto se ha cargado como una carga multiparte. Para obtener más información, consulte Carga y copia de objetos con la carga multiparte.
    - Replication status (Estado de replicación): el estado de replicación del objeto. Para obtener más información, consulte Obtención de información del estado de replicación.
    - Estado de cifrado: el cifrado del lado servidor usado para cifrar el objeto. Para obtener más información, consulte Protección de los datos con el cifrado del servidor.
    - Estado de clave de bucket: indica si una clave de bucket generada por AWS KMS se aplica al objeto. Para obtener más información, consulte Reducción del costo de SSE-KMS con las claves de bucket de Amazon S3.
    - Lista de control de acceso a objetos: una lista de control de acceso (ACL) para cada objeto que define a qué Cuentas de AWS o grupos se les concede acceso a este objeto y el tipo de acceso que se concede. Para obtener más información sobre este campo, consulte Uso del campo de objeto de ACL. Para obtener más información acerca de las ACL, consulte Información general de las Listas de control de acceso (ACL).
    - Propietario del objeto: el propietario del objeto.
    - Clase de almacenamiento: la clase de almacenamiento utilizada para almacenar el objeto.

- Agrupación por niveles inteligente: capas de acceso: indica la capa de acceso (frecuente o infrecuente) del objeto si estaba almacenado en la clase de almacenamiento S3 Intelligent-Tiering. Para obtener más información, consulte Clase de almacenamiento para optimizar automáticamente los datos con patrones de acceso cambiantes o desconocidos.
- ETag: la etiqueta de entidad (ETag) es un hash del objeto. La ETag solo refleja los cambios en el contenido de un objeto, no en sus metadatos. La ETag puede ser o no un resumen MD5 de los datos del objeto. Esto dependerá del método de creación del objeto y del tipo de cifrado. Para obtener más información, consulte Object en la Referencia de la API de Amazon Simple Storage Service.
- Algoritmo de suma de comprobación: indica el algoritmo utilizado para crear la suma de comprobación para el objeto.
- Todas las configuraciones de bloqueo de objetos: estado de bloqueo del objeto, incluidos los siguientes ajustes:
  - Bloqueo de objetos: modo de retención: grado de protección que se aplica al objeto, Gobernanza o Conformidad.
  - Bloqueo de objetos: fecha límite de retención: la fecha hasta la que no se puede eliminar un objeto bloqueado.
  - Bloqueo de objetos: estado de retención legal: estado de retención legal del objeto bloqueado.

  Para obtener más información acerca de Bloqueo de objetos de S3, consulte Cómo funciona Bloqueo de objetos de S3.

Para obtener más información acerca del contenido de un informe de inventario, consulte Lista de Amazon S3 Inventory.

Para obtener más información sobre cómo restringir el acceso a determinados campos de metadatos opcionales en una configuración de inventario, consulte Control de la creación de la configuración del informe de inventario de S3.

14. Seleccione Crear.

Uso de la API de REST para trabajar con el inventario de S3

A continuación, se indican las operaciones REST que puede utilizar para trabajar con Inventario de Amazon S3.

- DeleteBucketInventoryConfiguration
- GetBucketInventoryConfiguration
- ListBucketInventoryConfigurations
- PutBucketInventoryConfiguration

Configuración de notificaciones de eventos de Amazon S3 para completar el inventario

Puede configurar una notificación de evento de Amazon S3 para recibir un aviso cuando se crea el archivo de suma de comprobación del manifiesto, que indica que una lista de inventario se ha agregado al bucket de destino. El manifiesto es una lista actualizada de todas las listas de inventario en la ubicación de destino.

Amazon S3 puede publicar eventos en un tema de Amazon Simple Notification Service (Amazon SNS), una cola de Amazon Simple Queue Service (Amazon SQS) o una función de AWS Lambda. Para obtener más información, consulte Notificaciones de eventos de Amazon S3.

La siguiente configuración de notificación define que todos los archivos `manifest.checksum` agregados recientemente al bucket de destino se procesen con la función AWS Lambda de `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>
```

Para obtener más información, consulte Uso de AWS Lambda con Amazon S3 en la Guía para desarrolladores de AWS Lambda.

Localizar la descripción del inventario

Cuando se publica una lista de inventario, los archivos de manifiesto se publican en la siguiente ubicación del bucket de destino.

```
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* es el prefijo del nombre de la clave del objeto que se especifica opcionalmente en la configuración del inventario. Puede utilizar este prefijo para agrupar todos los archivos de lista de inventario en una ubicación común en el bucket de destino.
- *source-bucket* es el bucket de origen para el que se realiza la lista del inventario. El nombre del bucket de origen se añade para evitar colisiones cuando se envían varios informes de inventario procedentes de distintos buckets de origen al mismo bucket de destino.
- *config-ID* se añade para evitar las colisiones con varios informes de inventario del mismo bucket de origen que se envían al mismo bucket de destino. *config-ID* proviene de la configuración del informe de inventario, y es el nombre del informe que se define durante la configuración.
- *YYYY-MM-DDTHH-MMZ* es la marca temporal que consta de la hora y de la fecha de inicio en la que el proceso de generación del informe de inventario comienza a explorar el bucket; por ejemplo, `2016-11-06T21-32Z`.
- `manifest.json` es el archivo de manifiesto.
- `manifest.checksum` es el hash MD5 del contenido del archivo `manifest.json`.
- `symlink.txt` es el archivo de manifiesto compatible con Apache Hive.

Las listas de inventario se publican en la siguiente ubicación del bucket de destino a diario o cada semana.

```
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* es el prefijo del nombre de la clave del objeto que se especifica opcionalmente en la configuración del inventario. Puede utilizar este prefijo para agrupar todos los archivos de lista de inventario en una ubicación común en el bucket de destino.
- *source-bucket* es el bucket de origen para el que se realiza la lista del inventario. El nombre del bucket de origen se añade para evitar colisiones cuando se envían varios informes de inventario procedentes de distintos buckets de origen al mismo bucket de destino.
- *example-file-name*`.csv.gz` es uno de los archivos de inventario CSV. Los nombres de inventario ORC terminan con la extensión `.orc`, mientras que los nombres de inventario Parquet terminan con la extensión `.parquet`.

Manifiesto de inventario

Los archivos de manifiesto `manifest.json` y `symlink.txt` describen dónde se encuentran los archivos de inventario. Siempre que se entrega una nueva lista de inventario, esta va acompañada de un nuevo conjunto de archivos de manifiesto. Estos archivos pueden sobrescribirse entre sí. En buckets con control de versiones, Amazon S3 crea nuevas versiones de los archivos de manifiesto.

Cada manifiesto incluido en el archivo `manifest.json` proporciona metadatos y otra información básica sobre un inventario. La información incluye lo siguiente:

- Nombre del bucket de origen
- Nombre del bucket de destino
- Versión del inventario
- Marca temporal de creación en formato de fecha de inicio que consta de la hora y de la fecha de inicio en la que el proceso de generación del informe de inventario comienza a explorar el bucket
- Formato y esquema de los archivos de inventario
- Lista de los archivos de inventario que están en el bucket de destino

Cuando se escribe un archivo `manifest.json`, va acompañado de un archivo `manifest.checksum`, que es el hash MD5 del contenido del archivo `manifest.json`.

Example Manifiesto de inventario en un **`manifest.json`** archivo

En los ejemplos siguientes, se muestra un manifiesto de inventario en un archivo `manifest.json` para inventarios con formato CSV, ORC y Parquet.

CSV

A continuación se incluye un ejemplo de un manifiesto en un archivo `manifest.json` para un inventario con formato 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

A continuación se incluye un ejemplo de un manifiesto en un archivo `manifest.json` para un inventario con formato 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

A continuación, se incluye un ejemplo de un manifiesto en un archivo `manifest.json` para un inventario con formato 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"
        }
    ]
}
```

El archivo `symlink.txt` es un archivo de manifiesto compatible con Apache Hive que permite a Hive detectar automáticamente los archivos de inventario y archivos de datos asociados. El manifiesto compatible con Hive funciona con los servicios compatibles con Hive Athena y Amazon Redshift Spectrum. También funciona con aplicaciones compatibles con Hive, como Presto, Apache Hive, Apache Spark y muchas otras.

> **Important**
>
> El archivo de manifiesto `symlink.txt` compatible con Apache Hive no funciona actualmente con AWS Glue.
> El archivo `symlink.txt` no se puede leer con Apache Hive ni Apache Spark en los archivos de inventario con formato ORC o Parquet.

Consulta de Amazon S3 Inventory con Amazon Athena

Puede consultar archivos del Inventario de Amazon S3 con consultas SQL estándar utilizando Amazon Athena en todas las regiones donde Athena está disponible. Para verificar la disponibilidad de la Región de AWS, consulte la Tabla de Región de AWS.

Athena puede consultar los archivos de Inventario de Amazon S3 en el formato Optimized Row Columnar (ORC) de Apache, Apache Parquet o en el formato de valores separados por comas (CSV). Cuando se utiliza Athena para consultar archivos del inventario, es recomendable que se usen archivos de inventario con formato ORC o Parquet. Los formatos ORC y Parquet proporcionan mayor velocidad y menores costes de las consultas. ORC y Parquet son formatos de archivo ordenados en columnas autodescriptivos y con reconocimiento de tipos diseñados para Apache Hadoop. El formato en columnas permite al lector leer, descomprimir y procesar solo las columnas necesarias para la consulta actual. Los formatos ORC y Parquet del Inventario de Amazon S3 están disponibles en todas las Regiones de AWS.

Para usar Athena para consultar los archivos de Inventario de Amazon S3

1. Crear una tabla de Athena. Para obtener información sobre cómo crear una tabla, consulte Creación de tablas en Amazon Athena en la guía del usuario de Amazon Athena.
2. Cree su consulta mediante una de las siguientes plantillas de consulta de ejemplo, en función de si está consultando un informe de inventario con formato ORC, Parquet o CSV.

- Si utiliza Athena para consultar un informe de inventario con formato ORC, use la siguiente consulta de ejemplo como plantilla.

  La siguiente consulta de ejemplo incluye todos los campos opcionales del informe de inventario en formato ORC.

  Para utilizar esta consulta de ejemplo, haga lo siguiente:

  - Sustituya *your_table_name* por el nombre de la tabla de Athena que ha creado.
  - Elimine todos los campos opcionales que no haya seleccionado para su inventario para que la consulta se corresponda con los campos seleccionados para su inventario.
  - Reemplace el siguiente nombre de bucket y la ubicación de inventario (el ID de configuración) según corresponda a su configuración.

    `s3://amzn-s3-demo-bucket/config-ID/hive/`

  - Sustituya la fecha *2022-01-01-00-00* de `projection.dt.range` por el primer día del intervalo de tiempo dentro del cual va a particionar los datos en Athena. Para obtener más información, consulte Partitioning data in Athena (Particiones de datos en 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"
  );
```

- Si utiliza Athena para consultar un informe de inventario con formato Parquet, use la siguiente consulta de ejemplo para un informe con formato ORC. Sin embargo, utilice el siguiente SerDe de Parquet en lugar del SerDe de ORC en la instrucción de `ROW FORMAT SERDE`.

  ```
  ROW FORMAT SERDE 'org.apache.hadoop.hive.ql.io.parquet.serde.ParquetHiveSerDe'
  ```

- Si utiliza Athena para consultar un informe de inventario con formato CRV, use la siguiente consulta de ejemplo como plantilla.

  La siguiente consulta de ejemplo incluye todos los campos opcionales del informe de inventario en formato CSV.

  Para utilizar esta consulta de ejemplo, haga lo siguiente:

  - Sustituya *`your_table_name`* por el nombre de la tabla de Athena que ha creado.
  - Elimine todos los campos opcionales que no haya seleccionado para su inventario para que la consulta se corresponda con los campos seleccionados para su inventario.
  - Reemplace el siguiente nombre de bucket y la ubicación de inventario (el ID de configuración) según corresponda a su configuración.

    `s3://amzn-s3-demo-bucket/config-ID/hive/`

  - Sustituya la fecha *`2022-01-01-00-00`* de `projection.dt.range` por el primer día del intervalo de tiempo dentro del cual va a particionar los datos en Athena. Para obtener más información, consulte Partitioning data in Athena (Particiones de datos en 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. Ahora puede realizar varias consultas en su inventario, como se muestra en los siguientes ejemplos. Reemplace cada *user input placeholder* por su propia información.

```
# 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;
```

Al configurar el inventario de S3 para añadir el campo de la lista de control de acceso del objeto (ACL de objeto) a un informe de inventario, el informe muestra el valor del campo ACL del objeto como una cadena codificada en base64. Para obtener el valor descodificado en JSON para el campo ACL del objeto, puede consultar este campo en Athena. Vea los siguientes ejemplos de consultas. Para obtener más información acerca del campo ACL del objeto, consulte Uso del campo de objeto de ACL.

```
# 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;
```

Para obtener más información sobre el uso de Athena, consulte la Guía del usuario de Amazon Athena.

Convertir cadenas de ID de versión vacías de informes de Amazon S3 Inventory en cadenas nulas

> **Note**
>
> El siguiente procedimiento se aplica únicamente a los informes de Amazon S3 Inventory que incluyen todas las versiones, y solo si los informes de "todas las versiones" se utilizan como manifiestos para S3 Batch Operations en buckets que tienen habilitado S3 Versioning. No es necesario convertir cadenas para los informes de S3 Inventory que especifican únicamente la versión actual.

Puede utilizar informes de S3 Inventory como manifiestos para S3 Batch Operations. Sin embargo, cuando está habilitado S3 Versioning en un bucket, los informes de S3 Inventory que incluyen todas las versiones marcan cualquier objeto con versión nula con cadenas vacías en el campo ID de versión. Cuando un informe de inventario incluye todos los ID de versión de objeto, Batch Operations reconoce cadenas `null` como ID de versión, pero no cadenas vacías.

Cuando un trabajo de S3 Batch Operations utiliza un informe de S3 Inventory de "todas las versiones" como manifiesto, falla en todas las tareas de los objetos que tienen una cadena vacía en el campo ID de versión. Para convertir cadenas vacías en el campo ID de versión del informe de S3 Inventory en cadenas `null` para Batch Operations, utilice el siguiente procedimiento.

Actualice un informe de Amazon S3 Inventory para utilizarlo con Batch Operations

1. Inicie sesión en la AWS Management Console y abra la consola de Amazon S3 en https://console.aws.amazon.com/s3/.
2. Desplácese hasta el informe de S3 Inventory. El informe de inventario se encuentra en el bucket de destino que especificó al configurar el informe de inventario. Para obtener más

información sobre la localización de los informes de inventario, consulte Localizar la descripción del inventario.

a. Elija el del bucket de destino.

b. Elija la carpeta . La carpeta lleva el nombre original del bucket de origen.

c. Elija la carpeta que lleva el nombre de la configuración del inventario.

d. Seleccione la casilla de verificación que se encuentra junto a la carpeta denominada hive. En la parte superior de la página, elija Copy S3 URI (Copiar URI de S3) para copiar el URI de la carpeta.

3. Abra la consola de Amazon Athena en https://console.aws.amazon.com/athena/.

4. En el editor de consultas, elija Settings (Configuración) y, a continuación, Manage (Administrar). En la página Manage settings (Administrar la configuración), en Location of query result (Ubicación del resultado de la consulta), elija un bucket de S3 en el cual almacenar los resultados de la consulta.

5. En el editor de consultas, cree una tabla de Athena para almacenar los datos del informe de inventario mediante el siguiente comando. Reemplace *table_name* con el nombre de su elección, y en la cláusula `LOCATION`, inserte el URI de S3 que copió anteriormente. A continuación, elija Run (Ejecutar) para ejecutar la consulta.

```
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. Para borrar el editor de consultas, elija Clear (Borrar). A continuación, cargue el informe de inventario en la tabla mediante el siguiente comando. Reemplace *table_name* con el que eligió en el paso anterior. A continuación, elija Run (Ejecutar) para ejecutar la consulta.

```
MSCK REPAIR TABLE table_name;
```

7. Para borrar el editor de consultas, elija Clear (Borrar). Ejecute la siguiente consulta `SELECT` para recuperar todas las entradas del informe de inventario original y reemplazar los ID de versión vacíos por cadenas `null`. Reemplace *table_name* por el que eligió anteriormente y reemplace *YYYY-MM-DD-HH-MM* en la cláusula `WHERE` con la fecha del informe de inventario en la que desea que se ejecute esta herramienta. A continuación, elija Run (Ejecutar) para ejecutar la consulta.

```
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. Regrese a la consola de Amazon S3 (https://console.aws.amazon.com/s3/) y vaya al bucket de S3 que eligió anteriormente para Location of query result (Ubicación del resultado de la consulta). En el interior, debería haber una serie de carpetas que terminen con la fecha.

   Por ejemplo, debería ver algo similar a s3://***DOC-EXAMPLE-BUCKET***/***query-result-location***/Unsaved/2021/10/07/. Debería ver archivos `.csv` que contienen los resultados de la consulta `SELECT` que ejecutó.

   Elija el archivo CSV con la última fecha de modificación. Descargue este archivo en su equipo local para el siguiente paso.

9. El archivo CSV generado contiene una fila de encabezado. Para utilizar este archivo CSV como entrada para un trabajo de S3 Batch Operations, debe quitar la fila de encabezado, porque Batch Operations no admite filas de encabezado en los manifiestos CSV.

   Para eliminar la fila del encabezado, puede ejecutar uno de los siguientes comandos en el archivo. Reemplace *`file.csv`* con el nombre del archivo CSV.

   Para equipos macOS y Linux, ejecute el comando `tail` en una ventana de terminal.

   ```
   tail -n +2 file.csv > tmp.csv && mv tmp.csv file.csv
   ```

   Para máquinas Windows, ejecute el script siguiente en una ventana de Windows PowerShell. Sustituya *`File-location`* por la ruta a su archivo y *`file.csv`* por el nombre del archivo.

   ```
   $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. Después de eliminar la fila de encabezado del archivo CSV, podrá utilizarla como manifiesto en un trabajo de S3 Batch Operations. Cargue el archivo CSV en un bucket de S3 o ubicación que elija y, a continuación, cree un trabajo de Batch Operations utilizando el archivo CSV como manifiesto.

    Para obtener más información acerca de cómo crear un trabajo de Batch Operations, consulte Creación de trabajos de operaciones por lotes de S3.

Uso del campo de objeto de ACL

Un Inventario de Amazon S3 contiene una lista de los objetos del bucket de origen de S3 y los metadatos para cada objeto. El campo de lista de control de acceso (ACL) del objeto es un campo de metadatos que está disponible en el Inventario de Amazon S3. Específicamente, el campo ACL de objeto contiene la lista de control de acceso (ACL) de cada objeto. La ACL de un objeto define a qué Cuentas de AWS o grupos se les concede acceso a este objeto y el tipo de acceso que se concede. Para obtener más información, consulte Información general de las Listas de control de acceso (ACL) y Lista de Amazon S3 Inventory.

El campo ACL de objeto de los informes del Inventario de Amazon S3 se define en formato JSON. Los datos de JSON incluyen los siguientes campos:

- `version`: la versión del formato de campo ACL de objeto en los informes de inventario. Está en formato de fecha `yyyy-mm-dd`.
- `status`: los valores posibles son `AVAILABLE` o `UNAVAILABLE` para indicar si hay una ACL de objeto disponible para un objeto. Cuando el estado del campo ACL del objeto es `UNAVAILABLE`, el valor del campo Propietario del objeto del informe de inventario también es `UNAVAILABLE`.
- `grants`: pares de beneficiario-permiso que muestran el estado de cada beneficiario al que se le concede la ACL del objeto. Los valores disponibles para un beneficiario son `CanonicalUser` y `Group`. Para obtener más información sobre los beneficiarios, consulte la información sobre beneficiarios de las listas de control de acceso.

Para un beneficiario con el tipo `Group`, un par de beneficiario-permiso incluye los siguientes atributos:

- `uri`: un grupo predefinido de Amazon S3.
- `permission`: los permisos de ACL que se otorgan al objeto. Para obtener más información, consulte Permisos de ACL.
- `type`: el tipo `Group`, lo que indica que el beneficiario es un grupo.

Para un beneficiario con el tipo `CanonicalUser`, un par de beneficiario-permiso incluye los siguientes atributos:

- `canonicalId`: una forma oculta del ID de Cuenta de AWS. El ID de usuario canónico de una Cuenta de AWS es específico de esa cuenta. Puede recuperar el ID de usuario canónico. Para obtener más información, consulte Buscar el ID de usuario canónico de su Cuenta de AWS en la Guía de referencia de la Administración de cuentas de AWS.

  > **Note**
  >
  > Si el beneficiario de una ACL es la dirección de correo electrónico de una Cuenta de AWS, el Inventario de S3 utiliza el `canonicalId` de esa Cuenta de AWS y el tipo `CanonicalUser` para especificar este beneficiario. Para obtener más información, consulte la información sobre beneficiarios de las listas de control de acceso.

- `permission`: los permisos de ACL que se otorgan al objeto. Para obtener más información, consulte Permisos de ACL.
- `type`: el tipo `CanonicalUser`, lo que indica que el beneficiario es una Cuenta de AWS.

En el siguiente ejemplo, se muestran los valores posibles del campo ACL del objeto en formato 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**
>
> El campo ACL del objeto se define en formato JSON. Un informe de inventario muestra el valor del campo ACL del objeto como una cadena codificada en base64.
> Por ejemplo, supongamos que tiene el siguiente campo ACL de objeto en formato JSON:
>
> ```
> {
>         "version": "2022-11-10",
>         "status": "AVAILABLE",
>         "grants": [{
>             "canonicalId": "example-canonical-user-ID",
>             "type": "CanonicalUser",
>             "permission": "READ"
>         }]
> }
> ```
>
> El campo ACL del objeto está codificado y se muestra como la siguiente cadena codificada en base64:
>
> ```
> eyJ2ZXJzaW9uIjoiMjAyMi0xMS0xMCIsInN0YXR1cyI6IkFWQUlMQUJMRSIsImdyYW50cyI6W3siY2Fub25pY2FsSW
> ```
>
> Para obtener el valor descodificado en JSON para el campo ACL del objeto, puede consultar este campo en Amazon Athena. Para obtener consultas de ejemplo, consulte Consulta de Amazon S3 Inventory con Amazon Athena.

Información general de la replicación de objetos

Puede utilizar la replicación para habilitar la copia de objetos entre buckets de Amazon S3 de forma automática y asincrónica. Los buckets que están configurados para reproducción de objetos pueden pertenecer a la misma Cuenta de AWS o a cuentas diferentes. Puede replicar objetos en un solo bucket de destino o en varios buckets de destino. Los buckets de destino pueden estar en diferentes Regiones de AWS o dentro de la misma región que el bucket de origen.

Hay dos tipos de replicación: replicación en directo y replicación bajo demanda.

- Replicación en directo: para replicar automáticamente objetos nuevos y actualizados a medida que se escriben en el bucket de origen, utilice la replicación en directo. La replicación en directo no replica ningún objeto que hubiera en el bucket antes de configurar la replicación. Para replicar objetos que existían antes de configurar la replicación, utilice la replicación bajo demanda.
- Replicación bajo demanda: para replicar objetos existentes a partir del bucket de origen a uno o varios buckets de destino bajo demanda, utilice la replicación por lotes de S3. Para obtener más información sobre la replicación de objetos existentes, consulte Cuándo utilizar la replicación por lotes de S3.

Hay dos formas de replicación en directo: Replicación entre regiones (CRR) y Replicación de una sola región (SRR).

- Replicación entre regiones (CRR): se utiliza CRR para replicar objetos en buckets de S3 de diferentes Regiones de AWS. Para obtener más información acerca de CRR, consulte the section called “Cuándo utilizar la replicación entre regiones”.
- Replicación de una sola región (SRR): se utiliza SRR para copiar objetos en buckets de Amazon S3 de la misma Región de AWS. Para obtener más información sobre SRR, consulte the section called “Cuándo utilizar la replicación de la misma región”.

Temas

Motivos para usar la replicación

La replicación puede ayudarle a hacer lo siguiente:

- Replicar objetos reteniendo los metadatos: puede utilizar la replicación para realizar copias de los objetos en las que se retengan todos los metadatos, como las horas de creación del objeto original y los ID de versión. Esta capacidad es importante si debe asegurarse de que la réplica sea idéntica al objeto de origen.
- Replicar objetos en diferentes clases de almacenamiento: puede utilizar la replicación para colocar objetos directamente en S3 Glacier Flexible Retrieval, S3 Glacier Deep Archive u otra clase de almacenamiento en los buckets de destino. También puede replicar los datos en la misma clase de almacenamiento y utilizar las configuraciones de ciclo de vida en los buckets de destino para mover objetos a una clase de almacenamiento con menos actividad conforme adquieran antigüedad.
- Mantener copias de objetos con distintos propietarios: independientemente de quién sea el propietario del objeto de origen, puede indicar a Amazon S3 que cambie la propiedad de la réplica a la Cuenta de AWS que posee el bucket de destino. Esto se conoce como la opción de invalidación del propietario. Puede usar esta opción para restringir el acceso a las réplicas de objetos.
- Mantener los objetos almacenados en varias Regiones de AWS: para garantizar las diferencias geográficas en el lugar donde se guardan los datos, puede establecer varios buckets de destino en diferentes Regiones de AWS. Esta característica podría ayudarle a cumplir ciertos requisitos de conformidad.
- Replicar objetos en 15 minutos: para replicar sus datos en la misma Región de AWS o en distintas regiones dentro de un periodo predecible, puede utilizar el control de tiempo de replicación de S3 (S3 RTC). S3 RTC replica el 99,99 % de los objetos nuevos almacenados en Amazon S3) en un plazo de 15 minutos (con el respaldo de un acuerdo de nivel de servicio). Para obtener más información, consulte the section called “Uso del control de tiempo de replicación S3”.

> **Note**
>
> S3 RTC no se aplica a la replicación por lotes. La replicación por lotes es un trabajo de replicación bajo demanda y puede seguirse a través de la operación por lotes de S3. Para obtener más información, consulte Seguimiento del estado del trabajo e informes de finalización.

- Sincronizar buckets, replicar objetos existentes y replicar objetos que fallaron o se replicaron anteriormente: para sincronizar buckets y replicar objetos existentes, utilice la replicación por lotes como acción de replicación bajo demanda. Para obtener más información sobre cuándo utilizar la replicación por lotes, consulte Cuándo utilizar la replicación por lotes de S3.
- Replicar objetos y realizar una conmutación por error a un bucket en otra Región de AWS: para mantener todos los metadatos y objetos sincronizados entre los buckets durante la replicación de datos, utilice las reglas de replicación bidireccional (también conocidas como replicación bidireccional) antes de configurar los controles de conmutación por error de los puntos de acceso de varias regiones de Amazon S3. Las reglas de replicación bidireccional ayudan a garantizar que, cuando se escriben datos en el bucket de S3 al que se transfiere el tráfico por error, esos datos se repliquen de nuevo en el bucket de origen.

Cuándo utilizar la replicación entre regiones

La reproducción entre regiones (CRR) de S3 se utiliza para copiar objetos en buckets de Amazon S3 en diferentes Regiones de AWS. CRR puede ayudarlo a hacer lo siguiente:

- Cumplir los requisitos de conformidad: aunque Amazon S3 almacena sus datos en diversas zonas de disponibilidad alejadas geográficamente de forma predeterminada, los requisitos de conformidad pueden exigir que almacene los datos en ubicaciones aún más alejadas. Para cumplir con estos requisitos, utilice la replicación entre regiones para replicar los datos entre Regiones de AWS alejadas.
- Minimizar la latencia: si sus clientes están en dos ubicaciones geográficas, puede minimizar la latencia en el acceso a los objetos mediante el mantenimiento de copias de los objetos en Regiones de AWS que estén geográficamente más cerca de sus usuarios.
- Aumentar la eficiencia operativa: si tiene clústeres de computación en dos Regiones de AWS diferentes que analizan el mismo conjunto de objetos, puede optar por mantener copias de objetos en dichas regiones.

Cuándo utilizar la replicación de la misma región

La reproducción en la misma región (SRR) se utiliza para copiar objetos en buckets de Amazon S3 en la misma Región de AWS. SRR puede ayudarlo a hacer lo siguiente:

- Agregar registros en un solo bucket: si almacena registros en varios buckets o en varias cuentas, puede replicar registros fácilmente en un solo bucket en la región. Esto permite un procesamiento más simple de los registros en una sola ubicación.
- Configurar la replicación en directo entre las cuentas de producción y prueba: si usted o sus clientes tienen cuentas de producción y de prueba que utilizan los mismos datos, puede replicar objetos entre esas cuentas múltiples, mientras mantiene los metadatos de los objetos.
- Cumplir las leyes de soberanía de datos: es posible que tenga que almacenar varias copias de sus datos en Cuentas de AWS separadas dentro de una misma región. La replicación en la misma región puede ayudarle a replicar automáticamente los datos críticos cuando las normativas de conformidad no permitan que los datos salgan de su país.

Cuándo utilizar la replicación bidireccional (replicación bidireccional)

- Cree conjuntos de datos compartidos en varias Regiones de AWS: con la sincronización de modificaciones de réplicas, puede replicar fácilmente los cambios en los metadatos, como listas de control de acceso (ACL) de objetos, etiquetas de objetos o bloqueos de objetos en objetos de réplica. Esta replicación bidireccional es importante si desea mantener sincronizados todos los objetos y los cambios en los metadatos de los objetos. Puede habilitar la sincronización de modificaciones de réplicas en una regla de replicación nueva o existente al realizar una replicación bidireccional entre dos o más buckets de la misma o diferente Regiones de AWS.
- Mantenga los datos sincronizados en todas las regiones durante la conmutación por error: puede sincronizar los datos de los buckets entre Regiones de AWS configurando las reglas de replicación bidireccional con la replicación entre regiones (CRR) de S3 directamente desde un punto de acceso de varias regiones. Para tomar una decisión bien fundamentada sobre cuándo iniciar la conmutación por error, también puede habilitar las métricas de replicación de S3 para supervisar la replicación en Amazon CloudWatch, Control del tiempo de replicación de S3 (S3 RTC) o desde el punto de acceso multirregional.
- Haga que la aplicación tenga una alta disponibilidad: incluso en caso de que se produzca una interrupción del tráfico regional, puede utilizar reglas de replicación bidireccional para mantener todos los metadatos y objetos sincronizados en todos los buckets durante la replicación de datos.

Cuándo utilizar la replicación por lotes de S3

La replicación por lotes replica los objetos existentes en distintos buckets como opción bajo demanda. A diferencia de la replicación en directo, estos trabajos se pueden ejecutar según sea necesario. La replicación por lotes puede ayudarle a realizar lo siguiente:

- Replicar objetos existentes: puede utilizar la replicación por lotes para replicar objetos que se agregaron al bucket antes de configurar la replicación en la misma región o la replicación entre regiones.
- Replicar objetos que no se pudieron replicar anteriormente: puede filtrar un trabajo de replicación por lotes para intentar replicar objetos con un estado de replicación de FAILED (FALLIDO).
- Replicar objetos que ya se replicaron: es posible que tenga que almacenar varias copias de sus datos en Cuentas de AWS o Regiones de AWS separadas. La replicación por lotes puede replicar objetos existentes en destinos recién agregados.
- Replicar réplicas de objetos creados a partir de una regla de replicación: las configuraciones de replicación crean réplicas de objetos en buckets de destino. Las réplicas de objetos solo se pueden replicar con replicación por lotes.

Requisitos para las cargas de trabajo y la replicación en directo

Según los requisitos para la carga de trabajo, algunos tipos de replicación se adaptarán mejor a su caso de uso que otros. Utilice la siguiente tabla para determinar qué tipo de replicación debe utilizar para su situación y si debe utilizar la opción Control del tiempo de replicación de S3 (S3 RTC) para la carga de trabajo. S3 RTC replica el 99,99 % de los objetos nuevos almacenados en Amazon S3 en un plazo de 15 minutos (con el respaldo de un acuerdo de nivel de servicio o SLA). Para obtener más información, consulte the section called “Uso del control de tiempo de replicación S3”.

Comparación de los requisitos de cargas de trabajo para la replicación

Requisito de carga de trabajo	S3 RTC (SLA de 15 minutos)	Replicación entre regiones (CRR)	Replicación de una sola región (SRR)
Replicar objetos entre diferentes Cuentas de AWS	Sí	Sí	Sí
Replicar objetos en la misma Región de	No	No	Sí

Requisito de carga de trabajo	S3 RTC (SLA de 15 minutos)	Replicación entre regiones (CRR)	Replicación de una sola región (SRR)
AWS en un plazo de 24 a 48 horas (no está respaldado por un SLA)			
Replicar objetos entre diferentes Regiones de AWS en un plazo de 24 a 48 horas (no está respaldado por un SLA)	No	Sí	No
Tiempo de replicación predecible: respaldado por un SLA que permite replicar el 99,9 % de los objetos en 15 minutos	Sí	No	No

¿Qué replica Amazon S3?

Amazon S3 replica solo elementos específicos en buckets que están configurados para la replicación.

Temas

- ¿Qué se replica con las configuraciones de replicación?
- ¿Qué elementos no se replican con las configuraciones de replicación?
- Cómo afecta el cifrado de buckets predeterminado a la replicación

¿Qué se replica con las configuraciones de replicación?

De forma predeterminada, Amazon S3 replica lo siguiente:

- Objetos creados después de añadir una configuración de replicación.

- Objetos sin cifrar.
- Objetos cifrados mediante claves proporcionadas por el cliente (SSE-C), objetos cifrados en reposo bajo claves administradas de Amazon S3 (SSE-KMS) o una clave de KMS almacenada en AWS Key Management Service (SSE-KMS). Para obtener más información, consulte the section called “Replicar objetos cifrados”.
- Metadatos de objeto desde los objetos de origen hasta las réplicas. Para obtener información acerca de la replicación de metadatos de las réplicas a los objetos de origen, consulte Replicación de cambios de metadatos con la sincronización de modificación de réplica de Amazon S3.
- Solo los objetos en el bucket de origen para los que el propietario del bucket tiene permisos para leer objetos y listas de control de acceso (ACL).

  Para obtener más información acerca de la propiedad de recursos, consulte Propiedad de los buckets y objetos de Amazon S3.
- El objeto ACL se actualiza, a menos que ordene a Amazon S3 que cambie la propiedad de la réplica cuando los buckets de origen y destino no son propiedad de las mismas cuentas.

  Para obtener más información, consulte Cambiar el propietario de la réplica.

  Amazon S3 puede tardar tiempo en sincronizar las dos ACL. Este cambio de propiedad se aplica solo a los objetos creados luego de agregar una configuración de replicación al bucket.
- Etiquetas de objeto, si las hay.
- Información de retención de Bloqueo de objetos de S3, si la hay.

  Cuando Amazon S3 replica los objetos que tienen aplicada información de retención, aplica esos mismos controles de retención a las réplicas, lo que anula el periodo de retención predeterminado configurado en los buckets de destino. Si no tiene controles de retención aplicados a los objetos en el bucket de origen y se replican en buckets de destino que tienen establecido un periodo de retención predeterminado, el periodo de retención predeterminado del bucket de destino se aplica a las réplicas del objeto. Para obtener más información, consulte Usar Bloqueo de objetos de S3.

Cómo afectan las operaciones de eliminación a la replicación

Si elimina un objeto del bucket de origen, las siguientes acciones se producen de forma predeterminada:

- Si realiza una solicitud DELETE sin especificar un ID de versión del objeto, Amazon S3 añade un marcador de eliminación. Amazon S3 se ocupa del marcador de eliminación de la siguiente manera:
  - Si utiliza la última versión de la configuración de replicación (es decir, si especifica el `Filter` elemento en una regla de configuración de replicación), Amazon S3 no replicará el marcador de eliminación de forma predeterminada. Sin embargo, puede agregar la replicación de marcador de eliminación a reglas no basadas en etiquetas. Para obtener más información, consulte Replicación de marcadores de eliminación entre buckets.
  - Si no especifica el elemento `Filter`, Amazon S3 asume que la configuración de replicación es la versión V1 y replica los marcadores de eliminación resultantes de las acciones del usuario. Sin embargo, si Amazon S3 elimina un objeto debido a una acción de ciclo de vida, el marcador de eliminación no se replicará en los buckets de destino.
- Si especifica un ID de versión de objeto para eliminar en una solicitud `DELETE`, Amazon S3 elimina esa versión del objeto en el bucket de origen. Pero no replica la eliminación en el bucket de destino. En otras palabras, no elimina la misma versión del objeto de los buckets de destino. Esto protege los datos de eliminaciones malintencionadas.

¿Qué elementos no se replican con las configuraciones de replicación?

De forma predeterminada, Amazon S3 no replica lo siguiente:

- Los objetos en el bucket de origen que son réplicas, creadas por otra regla de replicación. Por ejemplo, imagine que configura la replicación donde el bucket A es el origen y el bucket B es el destino. Ahora, supongamos que añade otra configuración de replicación donde el bucket B es el de origen y el bucket C es el de destino. En este caso, los objetos en el bucket B que son réplicas de objetos en el bucket A no se replican en el bucket C.

  Para replicar objetos que son réplicas, utilice la replicación por lotes. Obtenga más información sobre cómo configurar la replicación por lotes en Replicación de objetos existentes.
- Objetos en el bucket de origen que ya se han replicado en un destino diferente. Por ejemplo, si cambia el bucket de destino en una configuración de replicación existente, Amazon S3 no replicará los objetos de nuevo.

  Para replicar objetos replicados anteriormente, utilice la replicación por lotes. Obtenga más información sobre cómo configurar la replicación por lotes en Replicación de objetos existentes.

- La replicación por lotes no admite volver a replicar objetos que se eliminaron con el ID de versión del objeto del bucket de destino. Para volver a replicar estos objetos, puede copiar los objetos de origen en su lugar con un trabajo de copia por lotes. Al copiar esos objetos en su lugar, se crean nuevas versiones de los objetos en el bucket de origen e inicia la replicación automáticamente en el destino. Para obtener más información acerca de cómo utilizar la copia por lotes, consulte Ejemplos donde se utilizan las operaciones por lotes para copiar objetos.
- De forma predeterminada, cuando replica desde una Cuenta de AWS diferente, los marcadores de eliminación agregados al bucket de origen no se replican.

  Para obtener más información sobre cómo replicar los marcadores de eliminación, consulte Replicación de marcadores de eliminación entre buckets.
- Los objetos almacenados en los niveles o clases de almacenamiento S3 Glacier Flexible Retrieval, S3 Glacier Deep Archive, S3 Intelligent-Tiering Archive Access o S3 Intelligent-Tiering Deep Archive Access. No puede replicar estos objetos hasta que los restaure y los copie en una clase de almacenamiento diferente.

  Para obtener más información sobre S3 Glacier Flexible Retrieval y S3 Glacier Deep Archive, consulte Clases de almacenamiento para objetos a los que se accede con poca frecuencia.

  Para obtener más información sobre S3 Intelligent-Tiering, consulte Amazon S3 Intelligent Tiering.
- Los objetos del bucket de origen para los que el propietario del bucket no tiene permisos suficientes de replicación.

  Para obtener información sobre cómo el propietario de un objeto puede conceder permisos al propietario de un bucket, consulte Conceder permisos entre cuentas para cargar objetos al mismo tiempo que se garantiza que el propietario del bucket tenga el control total.
- Actualiza a subrecursos de bucket.

  Por ejemplo, si cambia la configuración del ciclo de vida en una configuración de notificación al bucket de origen, estos cambios no se aplican al bucket de destino. Esta característica permite tener diferentes configuraciones en los buckets de origen y destino.
- Acciones realizadas por la configuración del ciclo de vida.

  Por ejemplo, si la configuración del ciclo de vida está habilitada solo en el bucket de origen, Amazon S3 crea marcadores de eliminación para los objetos que han vencido, pero no replica esos marcadores. Si desea que se aplique la misma configuración de ciclo de vida a los buckets de origen y destino, habilite la misma configuración de ciclo de vida en ambos. Para obtener más

información acerca de la configuración del ciclo de vida, consulte Administración del ciclo de vida del almacenamiento.

- Cuando utiliza reglas de replicación basadas en etiquetas con replicación en vivo, los objetos nuevos deben etiquetarse con la etiqueta de regla de replicación correspondiente en la operación `PutObject`. De lo contrario, los objetos no se replicarán. Si los objetos se etiquetan después de la operación `PutObject`, tampoco se replicarán.

  Para replicar objetos etiquetados después de la operación `PutObject`, debe utilizar la replicación por lotes de S3. Para obtener más información sobre la replicación por lotes, consulte Replicación de objetos existentes.

Cómo afecta el cifrado de buckets predeterminado a la replicación

Cuando habilita el cifrado predeterminado para un bucket de destino de replicación, se aplica el siguiente comportamiento de cifrado:

- Si los objetos del bucket de origen no están cifrados, los objetos de réplica del bucket de destino se cifran mediante la configuración de cifrado predeterminado del bucket de destino. Como resultado, las etiquetas de entidad (ETags) de los objetos de origen difieren de las ETags de los objetos de réplica. Si tiene aplicaciones que utilizan ETags, deberá actualizarlas para tener en cuenta esta diferencia.
- Si los objetos del bucket de origen se cifran mediante el cifrado del servidor con claves administradas por Amazon S3 (SSE-S3), el cifrado del servidor con claves de AWS Key Management Service (AWS KMS) (SSE-KMS) o con cifrado del servidor de doble capa con claves de AWS KMS (DSSE-KMS), los objetos de réplica del bucket de destino utilizarán el mismo tipo de cifrado que los objetos de origen. La configuración de cifrado predeterminado del bucket de destino no se utiliza.

Requisitos para la replicación y aspectos que hay que tener en cuenta

La replicación de Amazon S3 requiere lo siguiente:

- El propietario del bucket de origen debe tener habilitadas las Regiones de AWS de origen y destino en la cuenta. El propietario del bucket de destino debe tener la región de destino habilitada en la cuenta.

Para obtener más información sobre cómo habilitar o desactivar una Región de AWS, consulte Administración de Regiones de AWS en la Referencia general de AWS.

- Ambos buckets de origen y destino deben tener habilitado el control de versiones. Para obtener más información sobre el control de versiones, consulte Usar el control de versiones en buckets de S3.
- Amazon S3 debe tener permisos para replicar objetos desde el bucket de origen al bucket o buckets de destino en su nombre. Para obtener más información acerca de estos permisos, consulte Configuración de permisos para la replicación en directo.
- Si el propietario del bucket de origen no posee el objeto en el bucket, el propietario del objeto debe conceder al propietario del bucket los permisos `READ` y `READ_ACP` con la lista de control de acceso (ACL) del objeto. Para obtener más información, consulte Información general de las Listas de control de acceso (ACL).
- Si el bucket de origen tiene habilitado el bloqueo de objetos de S3, los buckets de destino deben tener también esa característica habilitada.

  Para habilitar la replicación en un bucket que tiene habilitado el bloqueo de objetos, debe usar la AWS Command Line Interface, la API de REST o los SDK de AWS. Para obtener más información general, consulte Usar Bloqueo de objetos de S3.

  > **Note**
  >
  > Tiene que conceder dos nuevos permisos en el bucket de S3 de origen del rol de AWS Identity and Access Management (IAM) que utiliza para configurar la reproducción. Los dos nuevos permisos son `s3:GetObjectRetention` y `s3:GetObjectLegalHold`. Si el rol tiene un permiso de `s3:Get*`, cumple el requisito. Para obtener más información, consulte Configuración de permisos para la replicación en directo.

Para obtener más información, consulte Configuración de la replicación en directo.

Si va a definir la configuración de reproducción en un escenario entre cuentas en el que los buckets de origen y destino pertenecen a diferentes Cuentas de AWS, se aplica el siguiente requisito:

- El propietario de los buckets de destino debe conceder al propietario del bucket de origen permisos para replicar objetos con una política de bucket. Para obtener más información, consulte

Concesión de permisos cuando los buckets de origen y destino son propiedad de diferentes Cuentas de AWS.

- Los buckets de destino no se pueden configurar como buckets de pago por solicitante. Para obtener más información, consulte Utilización de buckets de pago por solicitante para transferencias de almacenamiento y uso.

Aspectos que hay que tener en cuenta sobre la replicación

Antes de crear una configuración de replicación, tenga en cuenta lo siguiente.

Temas

Configuración de ciclo de vida y réplicas de objetos

El tiempo que Amazon S3 tarda en replicar un objeto depende del tamaño del objeto. Si los objetos son grandes, puede tardar varias horas. Aunque puede tardar un tiempo antes de que una réplica esté disponible en el destino, se tarda la misma cantidad de tiempo en crear la réplica que en crear el objeto correspondiente en el bucket de origen. Si una configuración de ciclo de vida está habilitada en un bucket de destino, las reglas del ciclo de vida respetan el tiempo de creación original del objeto, no en el momento en que la réplica estuvo disponible en el bucket de destino.

La configuración de replicación requiere que el bucket tenga habilitado el control de versiones. Cuando habilite el control de versiones en un bucket, tenga en cuenta lo siguiente:

- Si dispone de una configuración de ciclo de vida de vencimiento de objeto, después de habilitar el control de versiones, añada una política `NonCurrentVersionExpiration` para mantener el mismo comportamiento de eliminación permanente que antes de habilitar el control de versiones.
- Si tiene una configuración de ciclo de vida de transición, después de habilitar el control de versiones, considere la posibilidad de añadir la política `NonCurrentVersionTransition`.

Configuración del control de versión y la replicación

Cuando configura la replicación en un bucket, los buckets de origen y destino deben tener el control de versiones habilitado. Después de habilitar el control de versiones en los buckets de origen y destino, y de configurar la replicación en el bucket de origen, encontrará los siguientes problemas:

- Si intenta deshabilitar el control de versiones en el bucket de origen, Amazon S3 devolverá un error. Debe eliminar la configuración replicación antes de poder deshabilitar el control de versiones en el bucket de origen.
- Si deshabilita el control de versiones en el bucket de destino, la replicación generará un error. El objeto de origen tiene el estado de replicación `FAILED`.

Uso de la replicación de S3 con S3 Intelligent-Tiering

S3 Intelligent-Tiering es una clase de almacenamiento que está diseñada para optimizar los costes de almacenamiento moviendo automáticamente los datos a la capa de acceso más rentable. Por un pequeño cargo mensual de monitoreo y automatización de objetos, S3 Intelligent-Tiering monitorea los patrones de acceso y traslada automáticamente los objetos de una capa a otra.

La replicación de objetos almacenados en S3 Intelligent-Tiering con replicación por lotes de S3 o la invocación de CopyObject o UploadPartCopy se considera un acceso. En estos casos, los objetos de origen de las operaciones de copia o replicación están agrupados en niveles.

Para obtener más información acerca de S3 Intelligent-Tiering, consulte Amazon S3 Intelligent Tiering.

Configuración de registro y replicación

Si Amazon S3 distribuye registros a un bucket que tiene la replicación habilitada, replica los objetos del registro.

Si los registros de acceso del servidor (Registro de solicitudes con registro de acceso al servidor) o los registros de AWS CloudTrail (Registro de llamadas a la API de Amazon S3 mediante AWS CloudTrail) están habilitados en el bucket de origen o de destino, Amazon S3 incluye las solicitudes relacionadas con la replicación en los registros. Por ejemplo, Amazon S3 registra cada objeto que replica.

CRR y la región de destino

La replicación entre regiones (CRR) de Amazon S3 se utiliza para copiar objetos en buckets de S3 en diferentes Regiones de AWS. Puede elegir la región del bucket de destino en función de sus necesidades empresariales o consideraciones económicas. Por ejemplo, los cargos por transferencia de datos entre regiones varían en función de las regiones que elija.

Suponga que ha elegido EE. UU. Este (Norte de Virginia) (us-east-1) como la región para su bucket de origen. Si elige EE. UU. Oeste (Oregón) (us-west-2) como la región de los buckets de destino, pagará más que si elige la región EE. UU. Este (Ohio) (us-east-2). Para obtener información sobre los precios, consulte la sección "Precios de transferencia de datos" en Precios de Amazon S3.

No existen cargos por transferencia de datos asociados a la replicación entre regiones (SRR)

Replicación por lotes de S3

Para obtener información sobre los aspectos que hay que tener en cuenta para la replicación por lotes, consulte Consideraciones sobre la replicación por lotes de S3.

Control del tiempo de replicación de S3

Para obtener información sobre las prácticas recomendadas y los aspectos que hay que tener en cuenta para la opción Control del tiempo de replicación de S3 (S3 RTC), consulte Prácticas recomendadas y directrices para S3 RTC.

Configuración de la replicación en directo

> **Note**
>
> Los objetos que existían antes de configurar la replicación no se replican automáticamente. En otras palabras, Amazon S3 no replica los objetos retroactivamente. Para replicar objetos creados antes de la configuración de replicación, utilice la replicación por lotes de S3. Obtenga más información sobre cómo configurar la replicación por lotes en Replicación de objetos existentes.

Para habilitar la replicación en directo, (replicación en la misma región [SRR] o la replicación entre regiones [CRR]), añada una configuración de replicación al bucket de origen. Esta configuración indica a Amazon S3 que replique los objetos de la forma especificada. En la configuración de replicación, debe proporcionar lo siguiente:

- Los buckets de destino: el bucket o los buckets en los que desea que Amazon S3 replique los objetos.
- Los objetos que desea replicar: puede replicar todos los objetos del bucket de origen o de un subconjunto. Para identificar un subconjunto, proporcione un prefijo de nombre de clave, una o más etiquetas de objeto, o ambos en la configuración.

  Por ejemplo, si configura una regla de replicación para replicar solo objetos con el prefijo de nombre de clave `Tax/`, Amazon S3 replica objetos con claves como `Tax/doc1` o `Tax/doc2`. Pero no replica objetos con la clave `Legal/doc3`. Si especifica un prefijo y una o más etiquetas, Amazon S3 replica solo los objetos que tienen el prefijo de clave específico y las etiquetas.
- Un rol de AWS Identity and Access Management (IAM): Amazon S3 asume este rol de IAM para replicar objetos en su nombre.

Además de estos requisitos mínimos, puede elegir las siguientes opciones:

- Clase de almacenamiento de réplica: de forma predeterminada, Amazon S3 almacena réplicas de objetos mediante la misma clase de almacenamiento que el objeto de origen. Puede especificar una clase de almacenamiento diferente para las réplicas.
- Propiedad de la réplica: Amazon S3 supone que una réplica de objeto le pertenece al propietario del objeto de origen. Por tanto, cuando replica objetos, también replica la lista de control de acceso (ACL) del objeto correspondiente o configuración de S3 Object Ownership. Si los buckets de origen y destino pertenecen a dos Cuentas de AWS diferentes, puede configurar la reproducción para cambiar el propietario de una réplica a la Cuenta de AWS que posee el bucket de destino.

Puede configurar la replicación mediante la API de REST, los SDK de AWS, la AWS Command Line Interface (AWS CLI) o la consola de Amazon S3.

Amazon S3 también proporciona operaciones de API para que admita la configuración de reglas de replicación. Para obtener más información, consulte los siguientes temas en la referencia de la API de Amazon Simple Storage Service:

- PutBucketReplication
- GetBucketReplication
- DeleteBucketReplication

Temas

- Configuración de replicación
- Configuración de permisos para la replicación en directo
- Ejemplos para configurar la replicación en directo

Configuración de replicación

Amazon S3 almacena una configuración de replicación como XML. En el archivo XML de configuración de reproducción, usted especifica un rol de AWS Identity and Access Management (IAM) y una o más reglas.

```
<ReplicationConfiguration>
    <Role>IAM-role-ARN</Role>
    <Rule>
        ...
    </Rule>
    <Rule>
         ...
    </Rule>
     ...
</ReplicationConfiguration>
```

Amazon S3 no puede replicar objetos sin su permiso. Usted otorga permisos con el rol de IAM que especifique en la configuración de replicación. Amazon S3 asume el rol de IAM para replicar objetos en su nombre. Debe conceder primero los permisos necesarios al rol de IAM. Para obtener más información acerca de la administración de permisos, consulte Configuración de permisos para la replicación en directo.

Usted agrega una regla en una configuración de replicación en los siguientes casos:

- Desea replicar todos los objetos.
- Desea replicar un subconjunto de objetos. Identifica el subconjunto de objetos añadiendo un filtro en la regla. En el filtro, usted especifica un prefijo de clave de objeto, etiquetas o una combinación de ambos, para identificar el subconjunto de objetos a los que se aplica la regla. Los filtros se dirigen a los objetos que coinciden con los valores exactos que especifique.

Agrega varias reglas en una configuración de replicación si desea replicar un subconjunto diferente de objetos. En cada regla, se especifica un filtro que selecciona un subconjunto diferente de objetos. Por ejemplo, puede elegir replicar objetos que tengan los prefijos de clave `tax/` o `document/`. Para

ello, agregue dos reglas, una que especifique el filtro de prefijo de clave `tax/` y otro que especifique el prefijo de clave `document/`. Para obtener más información acerca de los prefijos de clave de objeto, consulte Organizar objetos con prefijos.

Las secciones siguientes facilitarán información adicional.

Temas

- Configuración básica de reglas
- Opcional: especificación de un filtro
- Configuraciones de destino adicionales
- Ejemplo de configuraciones de replicación
- Compatibilidad con versiones anteriores

Configuración básica de reglas

En cada regla se debe incluir el estado y la prioridad de la regla. Además, debe indicar si debe replicar marcadores de eliminación.

- `Status` indica si la regla está habilitada o desactivada mediante los valores `Enabled` o `Disabled`. Si una regla está desactivada, Amazon S3 no realiza las acciones especificadas en ella.
- `Priority` indica qué regla tiene prioridad cada vez que entran en conflicto dos o más reglas de replicación. Amazon S3 intenta replicar objetos de acuerdo con todas las reglas de replicación. Sin embargo, si hay dos o más reglas con el mismo bucket de destino, los objetos se replican de acuerdo a la regla con la prioridad más alta. Cuanto mayor sea el número, mayor será la prioridad.
- `DeleteMarkerReplication` indica si debe replicar marcadores de eliminación mediante los valores `Enabled` o `Disabled`.

En la configuración de destino, debe proporcionar el nombre del bucket o buckets donde desea que Amazon S3 replique los objetos.

En el siguiente ejemplo, se muestran los requisitos mínimos para una regla V2. Para la compatibilidad con versiones anteriores, Amazon S3 sigue siendo compatible con el formato XML V1. Para obtener más información, consulte Compatibilidad con versiones anteriores.

```
...
    <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:::amzn-s3-demo-bucket</Bucket>
        </Destination>
    </Rule>
    <Rule>
         ...
    </Rule>
     ...
...
```

También puede especificar otras opciones de configuración. Por ejemplo, puede elegir utilizar una clase de almacenamiento para réplicas de objetos que difieran de la clase para el objeto de origen.

Opcional: especificación de un filtro

Para elegir un subconjunto de objetos a los que se aplica la regla, añada un filtro opcional. Puede especificar un filtro utilizando un prefijo de clave de objeto, etiquetas de objeto o una combinación de ambos. Si filtra en un prefijo de clave y en etiquetas de objeto, Amazon S3 combina los filtros mediante un operador lógico AND. En otras palabras, la regla se aplica a un subconjunto de objetos con un prefijo de clave específico y etiquetas específicas.

Filtrar en función del prefijo de clave de objeto

Para especificar una regla con un filtro basado en un prefijo de la clave de un objeto, utilice el siguiente código. Puede especificar solo un prefijo.

```
<Rule>
    ...
    <Filter>
        <Prefix>key-prefix</Prefix>
    </Filter>
    ...
</Rule>
```

```
...
```

Filtrar en función de las etiquetas de objeto

Para especificar una regla con un filtro basado en etiquetas del objeto, utilice el siguiente código. También puede especificar una o varias etiquetas del objeto.

```
<Rule>
    ...
    <Filter>
        <And>
            <Tag>
                <Key>key1</Key>
                <Value>value1</Value>
            </Tag>
            <Tag>
                <Key>key2</Key>
                <Value>value2</Value>
            </Tag>
             ...
        </And>
    </Filter>
    ...
</Rule>
...
```

Filtrar con un prefijo de clave y etiquetas de objeto

Para especificar un filtro de reglas con una combinación de un prefijo de clave y etiquetas del objeto, use el código siguiente. Estos filtros se ajustan en un elemento <And> principal. Amazon S3 realiza una operación AND lógica para combinar estos filtros. En otras palabras, la regla se aplica a un subconjunto de objetos con un prefijo de clave específico y etiquetas específicas.

```
<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 especifica una regla con un elemento `<Filter>` vacío, la regla se aplica a todos los objetos del bucket.
> - Cuando utiliza reglas de replicación basadas en etiquetas con replicación en vivo, los objetos nuevos deben etiquetarse con la etiqueta de regla de replicación correspondiente en la operación `PutObject`. De lo contrario, los objetos no se replicarán. Si los objetos se etiquetan después de la operación `PutObject`, tampoco se replicarán.
>
> Para replicar objetos etiquetados después de la operación `PutObject`, debe utilizar la replicación por lotes de S3. Para obtener más información sobre la replicación por lotes, consulte Replicación de objetos existentes.

Configuraciones de destino adicionales

En la configuración de destino, especifique el bucket o los buckets en los que desea que Amazon S3 replique los objetos. Puede establecer configuraciones para replicar objetos de un bucket de origen a uno o más buckets de destino.

```
...
<Destination>
    <Bucket>arn:aws:s3:::amzn-s3-demo-bucket</Bucket>
</Destination>
...
```

Puede agregar las siguientes opciones en el `<Destination>` elemento.

Temas

- Especificar la clase de almacenamiento
- Agregar varios buckets de destino
- Especificar diferentes parámetros para cada regla de replicación con varios buckets de destino
- Cambiar la propiedad de la réplica
- Habilitar el control del tiempo de replicación de S3
- Replicar objetos creados con cifrado del lado del servidor mediante AWS KMS

Especificar la clase de almacenamiento

Puede especificar la clase de almacenamiento para las réplicas de objetos. De forma predeterminada, Amazon S3 utiliza la clase de almacenamiento del objeto de origen para crear réplicas de objetos, como en el ejemplo siguiente.

```
...
<Destination>
       <Bucket>arn:aws:s3:::amzn-s3-demo-bucket</Bucket>
       <StorageClass>storage-class</StorageClass>
</Destination>
...
```

Agregar varios buckets de destino

Puede agregar varios buckets de destino en una única configuración de replicación, como se indica a continuación.

```
...
<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>
...
```

Especificar diferentes parámetros para cada regla de replicación con varios buckets de destino

Al agregar varios buckets de destino en una única configuración de replicación, puede especificar parámetros diferentes para cada regla de replicación, como se indica a continuación.

```
...
<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>
...
```

Cambiar la propiedad de la réplica

Cuando los buckets de origen y destino no pertenecen a las mismas cuentas, puede cambiar el propietario de la réplica a la Cuenta de AWS que posee el bucket de destino. Para ello, agregue el elemento `AccessControlTranslation`. Este elemento toma el valor `Destination`.

```
...
<Destination>
   <Bucket>arn:aws:s3:::amzn-s3-demo-bucket</Bucket>
   <Account>destination-bucket-owner-account-id</Account>
   <AccessControlTranslation>
       <Owner>Destination</Owner>
   </AccessControlTranslation>
</Destination>
...
```

Si no agrega el elemento `AccessControlTranslation` a la configuración de replicación, las réplicas pertenecen a la misma Cuenta de AWS a la que pertenece el objeto de origen. Para obtener más información, consulte Cambiar el propietario de la réplica.

Habilitar el control del tiempo de replicación de S3

Puede habilitar S3 RTC (Control de tiempo de replicación de S3) en la configuración de replicación S3 RTC replica la mayoría de los objetos en pocos segundos y el 99,99 % de ellos en 15 minutos (con el respaldo de un acuerdo de nivel de servicio).

> **Note**
>
> Solo se acepta un valor de `<Minutes>15</Minutes>` para `EventThreshold` y `Time`.

```
...
<Destination>
  <Bucket>arn:aws:s3:::amzn-s3-demo-bucket</Bucket>
  <Metrics>
    <Status>Enabled</Status>
    <EventThreshold>
      <Minutes>15</Minutes>
    </EventThreshold>
  </Metrics>
  <ReplicationTime>
    <Status>Enabled</Status>
    <Time>
      <Minutes>15</Minutes>
    </Time>
  </ReplicationTime>
</Destination>
...
```

Para obtener más información, consulte Cumplimiento de los requisitos de conformidad mediante el control de tiempo de replicación de S3 (S3 RTC). Para conocer los ejemplos de la API, consulte PutBucketReplication en la referencia de la API de Amazon Simple Storage Service.

Replicar objetos creados con cifrado del lado del servidor mediante AWS KMS

Es posible que el bucket de origen contenga objetos creados con cifrado del lado del servidor mediante claves de AWS Key Management Service (AWS KMS) (SSE-KMS). De forma predeterminada, Amazon S3 no replica estos objetos. Opcionalmente, puede indicar a Amazon S3 que replique estos objetos. Para ello, primero opte explícitamente por esta característica agregando el elemento `SourceSelectionCriteria`. A continuación, proporcione la AWS KMS key (para la Región de AWS del bucket de destino) que se utilizará para cifrar réplicas de objetos. En el siguiente ejemplo, se muestra cómo especificar estos elementos.

```
...
<SourceSelectionCriteria>
  <SseKmsEncryptedObjects>
```

```
    <Status>Enabled</Status>
  </SseKmsEncryptedObjects>
</SourceSelectionCriteria>
<Destination>
  <Bucket>arn:aws:s3:::amzn-s3-demo-bucket</Bucket>
  <EncryptionConfiguration>
    <ReplicaKmsKeyID>AWS KMS key ID to use for encrypting object replicas</
ReplicaKmsKeyID>
  </EncryptionConfiguration>
</Destination>
...
```

Para obtener más información, consulte Replicación de objetos cifrados (SSE-C, SSE-S3, SSE-KMS, DSSE-KMS).

Ejemplo de configuraciones de replicación

Para comenzar, puede añadir los siguientes ejemplos de configuraciones de replicación a su bucket, según corresponda.

> **Important**
>
> Para añadir una configuración de replicación a un bucket, debe tener el permiso `iam:PassRole`. Este permiso le permite pasar el rol de IAM que otorga permisos de replicación de Amazon S3. Usted especifica el rol de IAM proporcionando el Nombre de recurso de Amazon (ARN) que se usa en el elemento `Role` en el XML de configuración de replicación. Para obtener más información, consulte Concesión de permisos a un usuario para transferir un rol a un servicio de Servicio de AWS en la Guía del usuario de IAM.

Example 1: configuración de replicación con una sola regla

La siguiente configuración de replicación básica especifica una regla. La regla especifica un rol de IAM que Amazon S3 puede asumir y un único bucket de destino para las réplicas de objetos. El valor `Status` de `Enabled` indica que la regla está en vigor.

```
<?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:::amzn-s3-demo-bucket</Bucket></Destination>

  </Rule>
</ReplicationConfiguration>
```

Para elegir un subconjunto de objetos para replicar, puede añadir un filtro. En la siguiente configuración, el filtro especifica un prefijo de clave de objeto. Esta regla se aplica a objetos que tienen el prefijo *Tax/* en sus nombres de clave.

```
<?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:::amzn-s3-demo-bucket</Bucket></Destination>

  </Rule>
</ReplicationConfiguration>
```

Si especifica el elemento `Filter`, también debe incluir los elementos `Priority` y `DeleteMarkerReplication`. En este ejemplo, `Priority` es irrelevante porque solo hay una regla.

En la siguiente configuración, el filtro especifica un prefijo y dos etiquetas. La regla se aplica al subconjunto de objetos que tengan el prefijo de clave y las etiquetas especificados. Específicamente, se aplica al objeto que tiene el prefijo *Tax/* en sus nombres de clave y las dos etiquetas de objetos especificadas. `Priority` no corresponde porque hay una sola regla.

```
<?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:::amzn-s3-demo-bucket</Bucket></Destination>

  </Rule>
</ReplicationConfiguration>
```

Puede especificar una clase de almacenamiento para las réplicas de objetos como se indica a continuación:

```
<?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:::amzn-s3-demo-bucket</Bucket>
       <StorageClass>storage-class</StorageClass>
```

```
    </Destination>
  </Rule>
</ReplicationConfiguration>
```

Puede especificar cualquier clase de almacenamiento compatible con Amazon S3.

Example 2: configuración de replicación con dos reglas

Example

En la siguiente configuración de replicación:

- Cada regla filtra un prefijo de clave diferente para que cada regla se aplique a un subconjunto distinto de objetos. En este ejemplo, Amazon S3 replica objetos con los nombres de clave *Tax/doc1.pdf* y *Project/project1.txt*, pero no replica ningún objeto con el nombre de clave *PersonalDoc/documentA*.
- La prioridad de regla es irrelevante porque las reglas se aplican a dos conjuntos de objetos distintos. El siguiente ejemplo muestra lo que ocurre cuando se aplica una prioridad de regla.
- En la segunda regla, se especifica la clase de almacenamiento S3 Standard-IA para las réplicas de objetos. Amazon S3 utiliza la clase de almacenamiento especificada para esas réplicas de objetos.

```
<?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: configuración de replicación con dos reglas con superposición de prefijos

En esta configuración, las dos reglas especifican filtros con superposición de prefijos de clave, *star/* y *starship/*. Ambas reglas se aplican a objetos con el nombre de clave *starship-x*. En este caso, Amazon S3 usará la prioridad de la regla para determinar la regla que se va a aplicar. Cuanto mayor sea el número, mayor será la prioridad.

```
<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: explicaciones de ejemplo

Para ver tutoriales de ejemplo, consulte Ejemplos para configurar la replicación en directo.

Para obtener más información acerca de la estructura de XML de configuración de replicación, consulte PutBucketReplication en la referencia de la API de Amazon Simple Storage Service.

Compatibilidad con versiones anteriores

La última versión del XML de la configuración de replicación es V2. Las configuraciones de replicación XML V2 son aquellas que contienen el elemento `Filter` para las reglas y las reglas que especifican S3 RTC (Control del tiempo de replicación de S3).

Para ver la versión de configuración de replicación, puede utilizar la operación de API `GetBucketReplication`. Para obtener más información, consulte GetBucketReplication en la Referencia de la API de Amazon Simple Storage Service.

Para la compatibilidad con versiones anteriores, Amazon S3 sigue siendo compatible con la configuración de replicación XML V1. Si ha utilizado la configuración de replicación XML V1, tenga en cuenta los siguientes problemas que afectan a la compatibilidad con versiones anteriores:

- El XML de configuración de replicación V2 incluye el elemento `Filter` para reglas. Con el elemento `Filter`, puede especificar filtros de objetos basados en el prefijo de la clave del objeto, etiquetas o ambos para abarcar los objetos a los que se aplica la regla. La configuración de réplica XML V1 admite filtrado basado solo en el prefijo de clave. En tal caso, añade el `Prefix` directamente como un elemento secundario del elemento `Rule`, como en el ejemplo siguiente.

```
<?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:::amzn-s3-demo-bucket</Bucket></Destination>

  </Rule>
</ReplicationConfiguration>
```

Para compatibilidad con versiones anteriores, Amazon S3 sigue siendo compatible con la configuración V1.

- Cuando elimina un objeto de su bucket de origen sin especificar un ID de versión del objeto, Amazon S3 agrega un marcador de eliminación. Si utiliza la configuración de replicación XML V1, Amazon S3 replica los marcadores de eliminación que resultan de las acciones de usuario. Dicho de otro modo, Amazon S3 replica el marcador de eliminación solo si un usuario elimina un objeto. Si Amazon S3 elimina un objeto caducado (como parte de una acción del ciclo de vida), Amazon S3 no replica el marcador de eliminación.

  En configuraciones de replicación V2, puede habilitar la replicación de marcadores de eliminación para reglas no basadas en etiquetas. Para obtener más información, consulte Replicación de marcadores de eliminación entre buckets.

Configuración de permisos para la replicación en directo

Cuando configura la replicación en directo, debe adquirir los permisos necesarios de la siguiente manera:

- Amazon S3 necesita permisos para replicar objetos en su nombre. Puede conceder estos permisos creando un rol de IAM y luego especificando ese rol en la configuración de replicación.
- Cuando los buckets de origen y destino no pertenecen a las mismas cuentas, el propietario del bucket de destino debe otorgar al propietario del bucket de origen permisos para almacenar las réplicas.

Temas

- Creación de un rol de IAM
- Concesión de permisos cuando los buckets de origen y destino son propiedad de diferentes Cuentas de AWS
- Granting permissions for S3 Batch Operations
- Cambio de la titularidad de la réplica
- Habilitar la recepción de objetos replicados desde un bucket de origen

Creación de un rol de IAM

De forma predeterminada, todos los recursos de Amazon S3 (buckets, objetos y subrecursos relacionados) son privados y solo el propietario del recurso puede acceder al él. Amazon S3 necesita permisos de lectura y replicación de objetos del bucket fuente. Puede conceder estos permisos creando un rol de IAM y especificar el rol en la configuración de replicación.

En esta sección se explica la política de confianza y la política de permisos mínimos necesarios. Los tutoriales de ejemplo proporcionan instrucciones paso a paso para crear un rol de IAM. Para obtener más información, consulte Ejemplos para configurar la replicación en directo.

- En el siguiente ejemplo, se muestra una política de confianza donde usted identifica a Amazon S3 como la entidad principal del servicio que puede asumir el rol.

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

- En el siguiente ejemplo, se muestra una política de confianza donde usted identifica a Amazon S3 como la entidad principal del servicio que puede asumir el rol. Esto resulta útil si está creando un trabajo de replicación por lotes. Para obtener más información, consulte Creación de un trabajo de replicación por lotes para una primera regla de replicación o un nuevo destino.

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

Para obtener más información acerca de los roles de IAM, consulte Roles de IAM en la guía del usuario de IAM.

- En el siguiente ejemplo, se muestra una política de acceso donde usted concede al rol permisos para realizar tareas de replicación en su nombre. Cuando Amazon S3 asume el rol, adopta los permisos que especifique en esta política. En esta política, *amzn-s3-demo-bucket1* es el bucket de origen y *amzn-s3-demo-bucket2* es el bucket de destino.

```
{
   "Version":"2012-10-17",
   "Statement":[
      {
         "Effect":"Allow",
         "Action":[
            "s3:GetReplicationConfiguration",
            "s3:ListBucket"
         ],
         "Resource":[
            "arn:aws:s3:::amzn-s3-demo-bucket1"
         ]
      },
      {
         "Effect":"Allow",
         "Action":[
            "s3:GetObjectVersionForReplication",
            "s3:GetObjectVersionAcl",
```

```
            "s3:GetObjectVersionTagging"
         ],
         "Resource":[
            "arn:aws:s3:::amzn-s3-demo-bucket1/*"
         ]
      },
      {
         "Effect":"Allow",
         "Action":[
            "s3:ReplicateObject",
            "s3:ReplicateDelete",
            "s3:ReplicateTags"
         ],
         "Resource":"arn:aws:s3:::amzn-s3-demo-bucket2/*"
      }
   ]
}
```

La política de acceso concede permisos para las siguientes acciones:

- `s3:GetReplicationConfiguration` y `s3:ListBucket`: Permisos para estas acciones en el bucket de *amzn-s3-demo-bucket1* (bucket de origen) permiten a Amazon S3 recuperar la configuración de replicación y mostrar el contenido del bucket. (El modelo de permisos actual requiere el permiso `s3:ListBucket` para acceder a marcadores de eliminación).
- `s3:GetObjectVersionForReplication` y `s3:GetObjectVersionAcl`: los permisos para estas acciones se conceden en todos los objetos para permitir que Amazon S3 obtenga una versión de objeto específica y una lista de control de acceso (ACL) asociada con los objetos.

- `s3:ReplicateObject` y `s3:ReplicateDelete`: los permisos para estas acciones en todos los objetos del bucket *amzn-s3-demo-bucket2* (el bucket de destino) permiten que Amazon S3 replique los objetos o marcadores de eliminación en el bucket de destino. Para obtener información acerca de los marcadores de eliminación, consulte Cómo afectan las operaciones de eliminación a la replicación.

> **Note**
>
> Los permisos para la acción `s3:ReplicateObject` en el bucket de *amzn-s3-demo-bucket2* (el bucket de destino) también permiten la replicación de metadatos, como las etiquetas de objetos y las ACLS. Por lo tanto, no es necesario que conceda permiso de forma explícita para la acción `s3:ReplicateTags`.

- `s3:GetObjectVersionTagging`: los permisos para esta acción en los objetos del bucket *`amzn-s3-demo-bucket1`* (el bucket de origen) permiten que Amazon S3 lea las etiquetas de objetos para la replicación. Para obtener más información, consulte Categorización del almacenamiento mediante etiquetas. Si Amazon S3 no tiene estos permisos, replica los objetos pero no las etiquetas de objetos.

Para obtener una lista de las acciones de Amazon S3, consulte Acciones, recursos y claves de condición para Amazon S3 en la Referencia de autorizaciones de servicios.

> **Important**
>
> La Cuenta de AWS propietaria del rol de IAM debe tener los permisos para las acciones que concede al rol de IAM.
> Por ejemplo, imagine que el bucket de origen contiene objetos que pertenecen a otra Cuenta de AWS. El propietario de los objetos debe conceder explícitamente a la Cuenta de AWS que posee el rol de IAM los permisos necesarios a través de la ACL del objeto. De lo contrario, Amazon S3 no puede acceder a los objetos y la replicación de los objetos dará un error. Para obtener más información acerca de los permisos de ACL, consulte Información general de las Listas de control de acceso (ACL).
> Los permisos aquí descritos están relacionados con la configuración de replicación mínima. Si elige agregar configuraciones de replicación opcionales, debe otorgar permisos adicionales a Amazon S3.

Concesión de permisos cuando los buckets de origen y destino son propiedad de diferentes Cuentas de AWS

Cuando los buckets de origen y destino no pertenecen a las mismas cuentas, el propietario del bucket de destino también debe agregar una política de bucket para conceder al propietario los permisos de propietario del bucket de origen con el objetivo de realizar las acciones de replicación de la siguiente manera: En esta política, *`amzn-s3-demo-bucket2`* es el bucket de destino.

> **Note**
>
> El formato de ARN del rol podría parecer diferente. Si el rol se crea mediante la consola, el formato de ARN es `arn:aws:iam::`*`account-ID`*`:role/service-role/`*`role-name`*.
> Si el rol se creó mediante la AWS CLI, el formato de ARN es `arn:aws:iam::`*`account-`*

ID:role/*role-name*. Para obtener más información, consulte Roles de IAM en la Guía del usuario de 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:::amzn-s3-demo-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::amzn-s3-demo-bucket2"
      }
   ]
}
```

Para ver un ejemplo, consulte La configuración de la reproducción para los buckets de origen y destino son propiedad de diferentes cuentas.

Si los objetos en el bucket de origen tienen etiquetas, tenga en cuenta lo siguiente:

- Si el propietario del bucket de origen concede permisos a Amazon S3 para las acciones `s3:GetObjectVersionTagging` y `s3:ReplicateTags` para replicar las etiquetas de los objetos (mediante el rol de IAM), Amazon S3 replicará las etiquetas junto con los objetos. Para obtener información acerca del rol de IAM, consulte Creación de un rol de IAM.
- Si el propietario del bucket de destino no desea replicar las etiquetas, puede añadir la siguiente instrucción a la política del bucket de destino para denegar el permiso explícitamente para la acción `s3:ReplicateTags`: En esta política, *amzn-s3-demo-bucket2* es el bucket de destino.

```
...
   "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:::amzn-s3-demo-bucket2/*"
      }
   ]
...
```

Granting permissions for S3 Batch Operations

La replicación por lotes de S3 proporciona una forma de replicar objetos que existían antes de que se estableciera una configuración de replicación, objetos que se han replicado anteriormente y objetos cuya replicación falló. Puede crear un trabajo de replicación por lotes único al crear la primera regla en una nueva configuración de replicación o al añadir un nuevo destino a una configuración existente a través de AWS Management Console. También puede iniciar la replicación por lotes para una configuración de replicación existente mediante la creación de un trabajo de operaciones por lotes.

Para ver ejemplos de políticas y roles de IAM de replicación por lotes, consulte Configuración de políticas de IAM para replicación por lotes.

Cambio de la titularidad de la réplica

Cuando las diferentes Cuentas de AWS son propietarias de los buckets de origen y destino, puede indicar a Amazon S3 que cambie la propiedad de la réplica a la Cuenta de AWS que posee el

bucket de destino. Para obtener más información sobre invalidar al propietario, consulte Cambiar el propietario de la réplica.

Habilitar la recepción de objetos replicados desde un bucket de origen

Puede generar rápidamente las políticas necesarias para permitir la recepción de objetos replicados desde un bucket de origen a través de AWS Management Console.

1. Inicie sesión AWS Management Console Management Console y abra la consola de Amazon S3 en https://console.aws.amazon.com/s3/.
2. En el panel de navegación izquierdo, elija Instancias.
3. En la lista Buckets, elija el bucket que desea utilizar como bucket de destino.
4. Elija la pestaña Management (Administración) y desplácese hacia abajo hasta Replication rules (Reglas de replicación).
5. Para Actions (Acciones), elija Receive replicated objects (Recibir objetos replicados).

   Siga las instrucciones e ingrese el ID de Cuenta de AWS de la cuenta del bucket de origen y elija Generate policies (Generar políticas). Esto generará una política de bucket de Amazon S3 y una política de claves de KMS.
6. Para agregar esta política a la política de bucket existente, elija Apply settings (Aplicar configuración) o elija Copy (Copiar) para copiar manualmente los cambios.
7. (Opcional) Copie la política de AWS KMS a la política de claves de KMS deseada en la consola de AWS Key Management Service.

Ejemplos para configurar la replicación en directo

Los siguientes ejemplos muestran cómo configurar la replicación en directo para casos de uso comunes.

> **Note**
>
> La replicación en directo hace referencia a la replicación en la misma región (SRR) y a la replicación entre regiones (CRR). La replicación en directo no replica ningún objeto que hubiera en el bucket antes de configurar la replicación. Para replicar objetos que existían antes de configurar la replicación, utilice la replicación bajo demanda. Para sincronizar buckets y replicar objetos existentes bajo demanda, consulte Replicación de objetos existentes.

En estos ejemplos se muestra cómo crear una configuración de replicación con la consola de Amazon S3, AWS Command Line Interface (AWS CLI) y los SDK de AWS (se muestran ejemplos de AWS SDK for Java y AWS SDK for .NET).

Para obtener más información acerca de cómo instalar y configurar la AWS CLI, consulte los siguientes temas en la Guía del usuario de la AWS Command Line Interface.

- Instalación de la AWS Command Line Interface
- Configuración de la AWS CLI: debe configurar al menos un perfil. Si está explorando escenarios que afectan a varias cuentas, configure dos perfiles.

Para obtener información sobre los SDK de AWS, consulte SDK de AWS para Java y SDK de AWS para .NET.

> **Tip**
>
> Para ver un tutorial paso a paso en el que se muestra cómo usar la replicación en directo para replicar datos, consulte el Tutorial: Replicación de datos dentro y entre Regiones de AWS mediante la replicación de S3.

Temas

- Configuración de la replicación para buckets de origen y destino que son propiedad de la misma cuenta
- La configuración de la reproducción para los buckets de origen y destino son propiedad de diferentes cuentas
- Cumplimiento de los requisitos de conformidad mediante el control de tiempo de replicación de S3 (S3 RTC)
- Replicación de objetos cifrados (SSE-C, SSE-S3, SSE-KMS, DSSE-KMS)
- Replicación de cambios de metadatos con la sincronización de modificación de réplica de Amazon S3
- Replicación de marcadores de eliminación entre buckets

Configuración de la replicación para buckets de origen y destino que son propiedad de la misma cuenta

La replicación consiste en la copia automática y asíncrona de los objetos de los buckets en las mismas o en diferentes Regiones de AWS. La replicación copia los objetos recientemente creados y las actualizaciones de objetos de un bucket de origen a un bucket o buckets de destino. Para obtener más información, consulte Información general de la replicación de objetos.

Al configurar la replicación, se agregan reglas de replicación al bucket de origen. Las reglas de replicación definen qué objetos del bucket de origen se deben replicar y el bucket o buckets de destino donde se almacenan los objetos replicados. Puede crear una regla para replicar todos los objetos en un bucket o un subconjunto de objetos con un prefijo de nombre de clave específico, una o varias etiquetas de objeto, o ambos métodos. El bucket de destino puede estar en la misma Cuenta de AWS que el bucket de origen o puede estar en una cuenta diferente.

Si especifica el ID de versión de objeto que desea eliminar, Amazon S3 elimina esa versión del objeto en el bucket de origen. Pero no replica la eliminación en el bucket de destino. En otras palabras, no elimina la misma versión del objeto del bucket de destino. Esto protege los datos de eliminaciones malintencionadas.

Cuando se añade una regla de replicación a un bucket, la regla está activada de forma predeterminada, por lo que comienza a funcionar tan pronto como se guarda.

En este ejemplo, configurará la replicación de los buckets de origen y destino que son propiedad de la misma Cuenta de AWS. Se proporcionan ejemplos de cómo utilizar la consola de Amazon S3, la AWS Command Line Interface (AWS CLI), y AWS SDK for Java y AWS SDK for .NET.

Uso de la consola de S3

Siga estos pasos para configurar una regla de replicación cuando el bucket de destino esté en la misma Cuenta de AWS que el bucket de origen.

Si el bucket de destino está en una cuenta distinta de la del bucket de origen, se debe añadir una política de bucket al bucket de destino para conceder al propietario de la cuenta del bucket de origen permiso para replicar objetos en el bucket de destino. Para obtener más información, consulte Concesión de permisos cuando los buckets de origen y destino son propiedad de diferentes Cuentas de AWS.

1. Inicie sesión AWS Management Console Management Console y abra la consola de Amazon S3 en https://console.aws.amazon.com/s3/.

2. En el panel de navegación izquierdo, elija Instancias.
3. En la lista Buckets, elija el nombre del bucket en cuestión.
4. Elija Administración, desplácese hacia abajo hasta Reglas de replicación y, a continuación, elija Crear regla de replicación.
5. En la sección Configuración de la regla de replicación en Nombre de la regla de replicación, introduzca un nombre para la regla para poder identificarla más adelante. El nombre es obligatorio y debe ser único dentro del bucket.
6. En Status (Estado), Enabled (Habilitada) está seleccionado de forma predeterminada. Una regla activada comienza a funcionar tan pronto se guarda. Si desea habilitar la regla más adelante, elija Deshabilitada.
7. Si el bucket tiene reglas de replicación existentes, se le indicará que establezca una prioridad para la regla. Debe establecer una prioridad para la regla para evitar conflictos causados por objetos incluidos en el ámbito de más de una regla. En caso de solaparse las reglas, Amazon S3 usará la prioridad de la regla para determinar la regla que se va a aplicar. Cuanto mayor sea el número, mayor será la prioridad. Para obtener más información acerca de la prioridad de regla, consulte Configuración de replicación.
8. En Bucket de origen, tiene las siguientes opciones para establecer el origen de la replicación:

   - Para replicar todo el bucket, elija Apply to all objects in the bucket (Aplicar a todos los objetos del bucket).
   - Para replicar todos los objetos que tengan mismo prefijo, elija Limit the scope of this rule using one or more filters (Limitar el ámbito de esta regla mediante uno o varios filtros). Esto limita la replicación a todos los objetos que tienen nombres que comienzan con el prefijo que ha especificado (por ejemplo, `pictures`). Introduzca un prefijo en el cuadro Prefijo.

     > **Note**
     >
     > Si utiliza un prefijo que es el nombre de una carpeta, debe introducir / (barra inclinada) como último carácter (por ejemplo, `pictures/`).

   - Para replicar todos los objetos con una o varias etiquetas de objeto, seleccione Agregar etiqueta y escriba el par clave-valor en los cuadros. Repita el procedimiento para añadir otra etiqueta. Puede hacer uso combinado de un prefijo y etiquetas. Para obtener más información acerca de las etiquetas de objeto, consulte Categorización del almacenamiento mediante etiquetas.

El nuevo esquema XML de configuración de replicación admite el filtrado por prefijos y etiquetas y la priorización de reglas. Para obtener más información acerca del nuevo esquema, consulte Compatibilidad con versiones anteriores. Para obtener más información sobre el XML utilizado con la API de Amazon S3 que funciona detrás de la interfaz de usuario, consulte Configuración de replicación. El nuevo esquema se describe como configuración de replicación XML V2.

9. En Destino, seleccione el bucket en el que desea que Amazon S3 replique los objetos.

> **Note**
>
> El número de buckets de destino está limitado al número de Regiones de AWS en una partición determinada. Una partición es una agrupación de regiones. AWS actualmente tiene tres particiones: `aws` (regiones estándar), `aws-cn` (regiones de China) y `aws-us-gov` (regiones de AWS GovCloud (US)). Puede usar cuotas de servicio para solicitar un aumento de la cuota del bucket de destino.

- Para replicar en un bucket o buckets de la cuenta, seleccione Elegir un bucket en esta cuenta y escriba o examine los nombres del bucket de destino.
- Para replicar en un bucket o buckets de una Cuenta de AWS diferente, seleccione Elegir un bucket de otra cuenta) e ingrese el ID de la cuenta del bucket de destino y el nombre del bucket.

  Si el destino está en una cuenta diferente de la del bucket de origen, debe agregar una política de bucket a los buckets de destino a fin de otorgar al propietario de la cuenta del bucket de origen permiso para replicar objetos. Para obtener más información, consulte Concesión de permisos cuando los buckets de origen y destino son propiedad de diferentes Cuentas de AWS.

  De forma opcional, si desea ayudar a estandarizar la propiedad de nuevos objetos en el bucket de destino, elija Cambiar la propiedad del objeto al propietario del bucket de destino. Para obtener más información acerca de esta opción, consulte Control de la propiedad de los objetos y desactivación de las ACL del bucket.

> **Note**
>
> Si el control de versiones no está habilitado en el bucket de destino, recibirá una advertencia que contiene un botón Enable versioning (Habilitar control de versiones). Elija este botón para activar el control de versiones en el bucket.

10. Configure un rol de AWS Identity and Access Management (IAM) que Amazon S3 pueda asumir para reproducir objetos en su nombre.

    Para configurar un rol de IAM, en la sección Rol de IAM, seleccione una de las opciones siguientes en la lista desplegable Rol de IAM:

    - Es absolutamente recomendable que elija Create new role (Crear nuevo rol) para que Amazon S3 cree un nuevo rol de IAM automáticamente. Cuando se guarda la regla, se genera una política nueva para el rol de IAM que coincide con los buckets de origen y de destino elegidos.
    - Puede elegir usar un rol de IAM existente. Si lo hace, debe elegir un rol que conceda a Amazon S3 los permisos necesarios para la replicación. La replicación dará un error si este rol no concede a Amazon S3 permisos suficientes para seguir la regla de replicación.

    > **Important**
    >
    > Cuando añada una regla de replicación a un bucket, debe tener el permiso `iam:PassRole` para poder pasar el rol de IAM que concede los permisos de replicación de Amazon S3. Para obtener más información, consulte Concesión de permisos a un usuario para transferir un rol a un Servicio de AWS en la Guía del usuario de IAM.

11. Para replicar objetos en el bucket de origen cifrados en el lado del servidor con claves de AWS Key Management Service (AWS KMS) (SSE-KMS), en Cifrado seleccione Replicar objetos cifrados con AWS KMS. En Claves de AWS KMS para cifrar objetos de destino están las claves de origen que permiten utilizar la replicación. De forma predeterminada, se incluyen todas las claves de KMS de origen. Para acotar la selección de claves de KMS puede seleccionar una alias o ID de clave.

    Los objetos cifrados por AWS KMS keys que no seleccione no se replican. Una clave KMS o un grupo de claves de KMS se seleccionan para usted, pero puede elegir las claves de KMS

que desee. Para obtener información acerca del uso de AWS KMS con replicación, consulte Replicación de objetos cifrados (SSE-C, SSE-S3, SSE-KMS, DSSE-KMS).

> **Important**
>
> Cuando replica objetos que están cifrados con AWS KMS, la tasa de solicitudes de AWS KMS se duplica en la región de origen y aumenta en la región de destino por la misma cantidad. Estas mayores tasas de llamada a AWS KMS se deben a la forma en la que los datos se vuelven a cifrar por medio de la clave de KMS que define para la región de destino de la replicación. AWS KMS tiene un cuota de tasa de solicitud por cada cuenta que realiza la llamada por cada región. Para obtener información acerca de los valores predeterminados de las cuotas, consulte Cuotas de AWS KMS - Solicitudes por segundo: varían en la Guía para desarrolladores de AWS Key Management Service.
> Si su tasa de solicitud actual de objeto PUT de Amazon S3 durante la replicación es más que la mitad del límite de la tasa de AWS KMS predeterminado para la cuenta, recomendamos que solicite un aumento de la cuaota de tasa de solicitud de AWS KMS. Para solicitar un aumento, abra un caso en el AWS Support Center en Contáctese con nosotros. Por ejemplo, suponga que su tasa de solicitud de objeto PUT actual es de 1000 solicitudes por segundo y utiliza AWS KMS para cifrar sus objetos. En ese caso, recomendamos que pida a AWS Support que aumente su límite de tasa de AWS KMS a 2500 solicitudes por segundo, tanto en la región de origen como en la de destino (si son diferentes), para garantizar que no haya una limitación controlada por AWS KMS.
> Para ver su tasa de solicitud de objeto PUT en el bucket de origen, consulte `PutRequests` en las métricas de solicitudes de Amazon CloudWatch para Amazon S3. Para obtener información sobre cómo ver las métricas de CloudWatch, consulte Uso de la consola de S3.

Si ha elegido los objetos replicados cifrados con AWS KMS, haga lo siguiente:

- En AWS KMS key para cifrar objetos de destino, especifique su clave de KMS de una de las siguientes maneras:
  - Para seleccionar de una lista de claves de KMS disponibles, marque Elija entre sus claves de AWS KMS keys y seleccione su clave de KMS en la lista de claves disponibles.

    En esta lista aparecen tanto la Clave administrada de AWS (`aws/s3`) como las claves administradas por el cliente. Para obtener más información acerca de las claves

administradas por el cliente, consulte Claves de cliente y claves de AWS en la Guía para desarrolladores de AWS Key Management Service.

- Para introducir el nombre de recurso de Amazon (ARN) de la clave de KMS, elija Introducir el ARN de AWS KMS key e introduzca el ARN de la clave de KMS en el campo que aparece. Esto cifra las réplicas en el bucket de destino. Puede buscar el ARN de su clave de KMS en la Consola de IAM bajo Claves de cifrado.
- Para crear una nueva clave administrada por el cliente en la consola de AWS KMS, elija Crear una clave de KMS.

  Para obtener más información acerca de cómo crear una AWS KMS key, consulte Creación de claves en la AWS Key Management Service Guía para desarrolladores.

> **Important**
>
> Solo puede utilizar las claves de KMS que estén habilitadas en la misma Región de AWS que el bucket. Cuando elige Elegir entre las claves de KMS, la consola de S3 solo muestra 100 claves de KMS por región. Si tiene más de 100 claves de KMS en la misma región, solo podrá ver las primeras 100 KMS en la consola S3. Para utilizar una clave de KMS que no aparece en la consola, elija Introducir el ARN de AWS KMS key y escriba el ARN de la clave de KMS.
> Cuando utilice una AWS KMS key para el cifrado en el lado del servidor en Amazon S3, debe elegir una clave de cifrado de KMS simétrica. Amazon S3 admite solo claves de KMS de cifrado simétricas y no claves de KMS asimétricas. Para obtener más información, consulte Identificación de claves de KMS simétricas y asimétricas en la Guía para desarrolladores de AWS Key Management Service.

Para obtener más información acerca de cómo crear una AWS KMS key, consulte Creación de claves en la AWS Key Management Service Guía para desarrolladores. Para obtener más información acerca del uso de AWS KMS con Amazon S3, consulte Uso del cifrado del servidor con claves de AWS KMS (SSE-KMS).

12. Si desea replicar los datos en una clase de almacenamiento específica del destino, en Clase de almacenamiento de destino, elija Cambiar la clase de almacenamiento para los objetos replicados. A continuación, elija la clase de almacenamiento que desea utilizar para los objetos replicados en el destino. Si no selecciona esta opción, la clase de almacenamiento de objetos replicados es la misma que la de los objetos originales.

13. Tiene las siguientes opciones adicionales al configurar las Opciones de replicación adicionales:

    - Si desea habilitar el Control del tiempo de replicación de S3 (S3 RTC) en la configuración de replicación, seleccione Control de tiempo de replicación (RTC). Para obtener más información acerca de esta opción, consulte Cumplimiento de los requisitos de conformidad mediante el control de tiempo de replicación de S3 (S3 RTC).
    - Si desea habilitar métricas de replicación de S3 en la configuración de replicación, seleccione Replication metrics and events (Métricas y eventos de replicación). Para obtener más información, consulte, Monitoreo del progreso con métricas de replicación y notificaciones de eventos de S3.
    - Si desea habilitar la replicación de marcador de eliminación en la configuración de replicación, seleccione Delete marker replication (Eliminar replicación de marcadores). Para obtener más información, consulte Replicación de marcadores de eliminación entre buckets.
    - Si desea habilitar la sincronización de modificación de réplica de Amazon S3 en la configuración de replicación, seleccione Replica modification sync (Sincronización de modificación de réplica). Para obtener más información, consulte, Replicación de cambios de metadatos con la sincronización de modificación de réplica de Amazon S3.

    > **Note**
    >
    > Cuando utiliza métricas de replicación S3 o S3 RTC, se aplican tarifas adicionales.

14. Para terminar, elija Save (Guardar).
15. Después de guardar la regla, puede seleccionar la regla y elegir Edit rule (Editar regla) para editarla, habilitarla, deshabilitarla o eliminarla.

Uso de la AWS CLI

Para utilizar la AWS CLI con el objetivo de configurar la replicación cuando los buckets de origen y destino son propiedad de la misma Cuenta de AWS, debe hacer lo siguiente:

- crear buckets de origen y destino
- habilitar el control de versiones en un bucket
- crear un rol de IAM que conceda permisos para replicar objetos en Amazon S)
- agregar la configuración de replicación al bucket de origen

Para verificar la configuración, debe probarla.

Para configurar la replicación cuando los buckets de origen y destino son propiedad de la misma Cuenta de AWS

1. Configure un perfil de credenciales para la AWS CLI. En este ejemplo, usamos el nombre de perfil `acctA`. Para obtener información acerca de la configuración de perfiles de credenciales, consulte Perfiles con nombre en la Guía del usuario de la AWS Command Line Interface.

   > **Important**
   >
   > El perfil utilizado para este ejercicio tiene que tener los permisos necesarios. Por ejemplo, en la configuración de replicación especifica el rol de IAM que Amazon S3 puede asumir. Solo puede hacer esto si el perfil que utiliza tiene el permiso `iam:PassRole`. Para obtener más información, consulte Concesión de permisos a un usuario para transferir un rol a un servicio de AWS en la Guía del usuario de IAM. Si utiliza credenciales de administrador para crear un perfil con nombre, puede realizar todas las tareas.

2. Cree un bucket *`source`* y habilite el control de versiones. El siguiente código crea un bucket *`source`* en la región Este de EE. UU. (Norte de Virginia) (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. Cree un bucket *`destination`* y habilite el control de versiones. El siguiente código crea un bucket *`destination`* en la región Oeste de EE. UU. (Oregón) (us-west-2).

> **Note**
>
> Para establecer la configuración de replicación cuando los buckets de origen y destino están en la misma Cuenta de AWS, debe utilizar el mismo perfil. En este ejemplo se utiliza `acctA`. Para probar la configuración de replicación cuando los buckets son propiedad de diferentes Cuentas de AWS, debe especificar diferentes perfiles para cada uno. En este ejemplo se usa el perfil `acctB` para el bucket de destino.

```
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. Cree un rol de IAM. Especifique este rol en la configuración de replicación que agregue al bucket de *origen* más adelante. Amazon S3 asume este rol para replicar objetos en su nombre. Crea el rol de IAM en dos pasos:

   - Crear un rol.
   - Asocie una política de permisos al rol.

   a. Cree el rol de IAM.

      i. Copie la siguiente política de confianza y guárdela en un archivo llamado `s3-role-trust-policy.json` en el directorio actual en su equipo local. Esta política concede a la entidad principal de servicio de Amazon S3 permisos para asumir el rol.

```
{
   "Version":"2012-10-17",
   "Statement":[
      {
```

```
         "Effect":"Allow",
         "Principal":{
            "Service":"s3.amazonaws.com"
         },
         "Action":"sts:AssumeRole"
      }
   ]
}
```

ii. Ejecute el siguiente comando para crear un rol.

```
$ aws iam create-role \
--role-name replicationRole \
--assume-role-policy-document file://s3-role-trust-policy.json  \
--profile acctA
```

b. Asocie una política de permisos al rol.

i. Copie la siguiente política de permisos y guárdela en un archivo llamado `s3-role-permissions-policy.json` en el directorio actual en su equipo local. Esta política concede permisos para varias acciones de buckets y objetos de 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. Ejecute el siguiente comando para crear una política y asociarla al rol.

```
$ aws iam put-role-policy \
--role-name replicationRole \
--policy-document file://s3-role-permissions-policy.json \
--policy-name replicationRolePolicy \
--profile acctA
```

5. Agregue la configuración de replicación al bucket *source*.

   a. Si bien la API de Amazon S3 requiere la configuración de replicación como XML, la AWS CLI requiere que especifique la configuración de reproducción como JSON. Guarde la siguiente JSON en un archivo denominado `replication.json` en el directorio local en su equipo.

```
{
  "Role": "IAM-role-ARN",
  "Rules": [
    {
      "Status": "Enabled",
      "Priority": 1,
      "DeleteMarkerReplication": { "Status": "Disabled" },
      "Filter" : { "Prefix": "Tax"},
      "Destination": {
        "Bucket": "arn:aws:s3:::destination-bucket"
      }
    }
  ]
```

```
}
```

b. Actualice JSON proporcionando valores para la *destination-bucket* y *IAM-role-ARN*. Guarde los cambios.

c. Ejecute el siguiente comando para añadir la configuración de replicación al bucket de origen. Asegúrese de proporcionar el nombre del bucket *source*.

```
$ aws s3api put-bucket-replication \
--replication-configuration file://replication.json \
--bucket source \
--profile acctA
```

Para recuperar la configuración de replicación, utilice el comando `get-bucket-replication`.

```
$ aws s3api get-bucket-replication \
--bucket source \
--profile acctA
```

6. Pruebe la configuración en la consola de Amazon S3:

a. Inicie sesión AWS Management Console Management Console y abra la consola de Amazon S3 en https://console.aws.amazon.com/s3/.

b. En el bucket *source*, cree una carpeta llamada `Tax`.

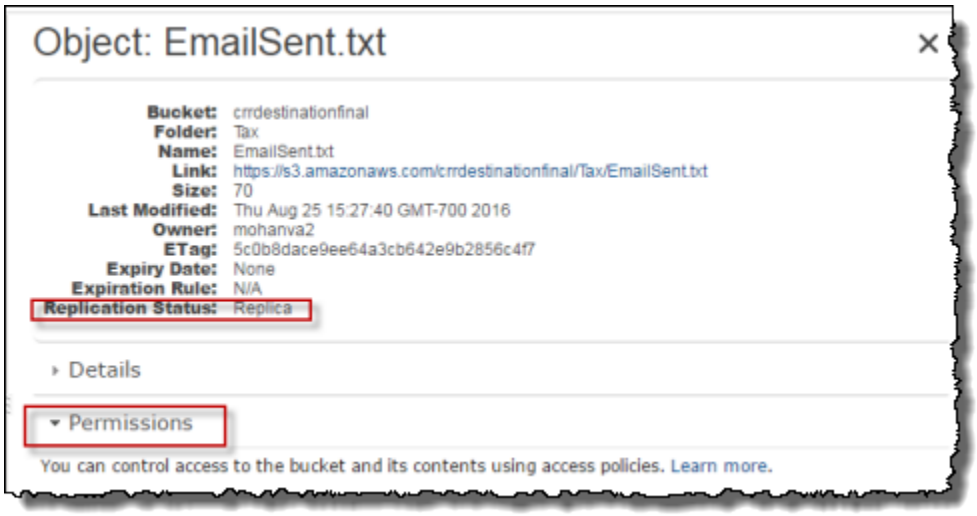
c. Agregue objetos de ejemplo a la carpeta `Tax` en el bucket *source*.

> **Note**
>
> La cantidad de tiempo que Amazon S3 tarda en replicar un objeto depende del tamaño del objeto. Para obtener más información sobre cómo ver el estado de la replicación, consulte Obtención de información del estado de replicación.

En el bucket *destination*, compruebe lo siguiente:

- Que Amazon S3 haya replicado los objetos.
- En las properties (propiedades) del objeto, que Replication Status (Estado de replicación) está establecido en `Replica` (lo que lo identifica como un objeto de réplica).

- En las propiedades del objeto, que la sección de permisos no muestra ningún permiso. Esto significa que la réplica aún pertenece al propietario del bucket *source* y que el propietario del bucket *destination* no tiene permisos en la réplica del objeto. Puede agregar opciones de configuración adicionales para indicar a Amazon S3 que cambie la titularidad de la réplica. Para ver un ejemplo, consulte Cómo cambiar al propietario de la réplica.



Uso de los AWS SDK

Utilice los siguientes ejemplos de código para añadir una configuración de replicación a un bucket con AWS SDK for Java y AWS SDK for .NET, respectivamente.

Java

El siguiente ejemplo añade una configuración de replicación a un bucket y luego recupera y verifica la configuración. Para obtener instrucciones sobre cómo crear y probar una muestra funcional, consulte Introducción en la Guía del desarrollador de AWS SDK for Java.

```
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#

El siguiente código de ejemplo de AWS SDK for .NET añade una configuración de replicación a un bucket y luego la recupera. Para usar este código, proporcione los nombres de los buckets y el nombre de recurso de Amazon (ARN) para el rol de IAM. Para obtener información acerca de cómo configurar y ejecutar ejemplos de código, consulte Introducción al SDK de AWS para .NET en la Guía para desarrolladores del SDK de AWS para .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);
            }
        }
```

```
    }
}
```

La configuración de la reproducción para los buckets de origen y destino son propiedad de diferentes cuentas

La configuración de replicación cuando los buckets de *origen* y *destino* son propiedad de diferentes Cuentas de AWS es similar a la configuración de replicación cuando los dos buckets son propiedad de la misma cuenta. La única diferencia es que el propietario del bucket de *destino* debe otorgar al propietario del bucket de *origen* permiso para replicar objetos añadiendo una política de bucket.

Para obtener más información acerca de la configuración de la replicación mediante el cifrado del lado del servidor con AWS Key Management Service en escenarios entre cuentas, consulte Conceder permisos adicionales para escenarios que afectan a varias cuentas.

Para configurar la replicación cuando los buckets de origen y destino son propiedad de diferentes Cuentas de AWS

1. En este ejemplo, crea los buckets de *origen* y *destino* en dos Cuentas de AWS diferentes. Debe tener dos perfiles de credenciales configurados para la AWS CLI (en este ejemplo, utilizamos `acctA` y `acctB` para los nombres de perfil). Para obtener más información acerca de la configuración de perfiles de credenciales, consulte Perfiles con nombre en la Guía del usuario de AWS Command Line Interface.
2. Siga las instrucciones paso a paso en Configuración de buckets en la misma cuenta con los siguientes cambios:
   - Para todos los comandos de la AWS CLI relacionados con actividades del bucket de *origen* (para crear el bucket de *origen*, habilitar el control de versiones y crear el rol de IAM), utilice el perfil `acctA`. Utilice el perfil `acctB` para crear el bucket de *destino*.
   - Asegúrese de que la política de permisos especifica los bucket de *origen* y de *destino* creados para este ejemplo.
3. En la consola, añada la siguiente política de bucket al bucket de *destino* para permitir al propietario del bucket de *origen* replicar objetos: Asegúrese de editar la política proporcionando el ID de la Cuenta de AWS del propietario del bucket de *origen* y el nombre del bucket de *destino*.

> **Note**
>
> Para utilizar el ejemplo siguiente, sustituya *user input placeholders* con su propia información. Sustituya *DOC-EXAMPLE-BUCKET* por el nombre de su bucket de destino. Sustituya *source-bucket-acct-ID:role/service-role/source-acct-IAM-role* por el rol que está utilizando para esta configuración de replicación.
> Si creó el rol de servicio de IAM manualmente, establezca la ruta del rol como `role/service-role/`, tal y como se muestra en el siguiente ejemplo de política. Para obtener más información, consulte ARN de IAM en la guía del usuario de 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"
      }
   ]
}
```

Elija el bucket y añada la política de buckets. Para obtener instrucciones, consulte Agregar una política de bucket mediante la consola de Amazon S3.

En la replicación, el propietario del objeto de origen es propietario de la réplica de manera predeterminada. Cuando los buckets de origen y destino son propiedad de diferentes Cuentas de AWS, puede agregar ajustes de configuración opcionales para cambiar la propiedad de la réplica a la Cuenta de AWS propietaria de los buckets de destino. Esto incluye la concesión del permiso `ObjectOwnerOverrideToBucketOwner`. Para obtener más información, consulte Cambiar el propietario de la réplica.

Cambiar el propietario de la réplica

En la replicación, el propietario del objeto de origen también es propietario de la réplica de manera predeterminada. Cuando los buckets de origen y destino pertenecen a diferentes Cuentas de AWS y desea cambiar la propiedad de réplica a la Cuenta de AWS que posee el bucket de destino, puede agregar ajustes de configuración opcionales para cambiar la propiedad de la réplica a la Cuenta de AWS propietaria de los buckets de destino. Podría hacer esto, por ejemplo, para restringir el acceso a las réplicas de objetos. Esta opción recibe el nombre también de invalidación del propietario en la configuración de replicación. Para obtener más información sobre la opción de invalidación del propietario, consulte Agregar la opción de invalidación del propietario a la configuración de la replicación. Para obtener información acerca de la configuración de replicación, consulte Información general de la replicación de objetos.

Para configurar la invalidación del propietario, haga lo siguiente:

- Añada la opción de invalidación del propietario a la configuración de replicación para indicar a Amazon S3 que cambie la titularidad de la réplica.
- Conceda permisos a Amazon S3 para cambiar la titularidad de la réplica.
- Agregue permisos en la política del buckets de destino para permitir cambiar la propiedad de la réplica. Esto permite al propietario de los buckets de destino aceptar la propiedad de las réplicas de objetos.

Para obtener más información, consulte Agregar la opción de invalidación del propietario a la configuración de la replicación. Para ver un ejemplo práctico con instrucciones paso a paso, consulte Cómo cambiar al propietario de la réplica.

Configuración de propietario del bucket obligatorio de Object Ownership

Cuando utiliza la replicación de Amazon S3 y los buckets de origen y destino pertenecen a diferentes Cuentas de AWS, el propietario del bucket de destino puede desactivar las ACL (con la configuración de propietario del bucket obligatorio para Object Ownership) para cambiar la propiedad de réplica a la Cuenta de AWS que posee el bucket de destino. Esta configuración imita el comportamiento de anulación del propietario existente sin necesidad del permiso `s3:ObjectOwnerOverrideToBucketOwner`. Esto significa que todos los objetos que se replican en el bucket de destino con la configuración de propietario del bucket obligatorio pertenecen al propietario del bucket de destino. Para obtener más información acerca de la propiedad de objetos, consulte Control de la propiedad de los objetos y desactivación de las ACL del bucket.

Agregar la opción de invalidación del propietario a la configuración de la replicación

> **Warning**
>
> Agregue la opción de invalidación del propietario solo cuando los buckets de origen y destino pertenezcan a distintas Cuentas de AWS. Amazon S3 no comprueba si los buckets pertenecen a las mismas cuentas o a cuentas diferentes. Si agrega la opción de invalidación del propietario cuando ambos buckets pertenecen a la misma Cuenta de AWS, Amazon S3 aplica la opción de invalidación del propietario. Concede permisos completos al propietario del bucket de destino y no replica las actualizaciones que se realicen posteriormente en la lista de control de acceso (ACL) del objeto de origen. El propietario de la réplica puede cambiar directamente la ACL asociada a una réplica con una solicitud `PUT ACL`, pero no a través de la replicación.

Para especificar la opción de sustitución de propietario, agregue lo siguiente a cada `Destination` elemento:

- El elemento `AccessControlTranslation`, que indica a Amazon S3 que cambie la titularidad de la réplica.
- El elemento `Account`, que especifica la Cuenta de AWS del propietario del bucket de destino.

```
<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>
```

La siguiente configuración de replicación de ejemplo, indica a Amazon S3 que replique objetos que tiene el prefijo de clave `Tax` en el bucket de destino y cambie la propiedad de las 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>
```

Concesión de permisos a Amazon S3 para cambiar la titularidad de la réplica

Conceda a Amazon S3 permisos para cambiar la titularidad de la réplica añadiendo permiso para la acción `s3:ObjectOwnerOverrideToBucketOwner` en la política de permisos asociada con el rol de IAM. Se trata del rol de IAM que especificó en la configuración de replicación que permite a Amazon S3 asumir y replicar objetos en su nombre.

```
...
{
    "Effect":"Allow",
```

```
          "Action":[
        "s3:ObjectOwnerOverrideToBucketOwner"
    ],
    "Resource":"arn:aws:s3:::destination-bucket/*"
}
...
```

Agregar permiso a la política del bucket de destino para permitir cambiar la titularidad de la réplica

El propietario del bucket de destino debe otorgar al propietario del bucket de origen permiso para cambiar la titularidad de la réplica. El propietario del bucket de destino otorga al propietario del bucket de origen permiso para la acción `s3:ObjectOwnerOverrideToBucketOwner`. Esto permite al propietario del bucket de destino aceptar la propiedad de las réplicas de objetos. En el siguiente ejemplo de instrucción de política de buckets se muestra cómo se hace esto.

```
...
{
    "Sid":"1",
    "Effect":"Allow",
    "Principal":{"AWS":"source-bucket-account-id"},
    "Action":["s3:ObjectOwnerOverrideToBucketOwner"],
    "Resource":"arn:aws:s3:::destination-bucket/*"
}
...
```

Consideraciones adicionales

Al configurar la opción de anulación de propiedad, se aplican las siguientes consideraciones:

- De forma predeterminada, el propietario del objeto de origen también es propietario de la réplica. Amazon S3 replica la versión del objeto y la ACL asociada a ella.

  Si añade la invalidación del propietario, Amazon S3 replica solo la versión del objeto, no la ACL. Además, Amazon S3 no replica los cambios que se realicen posteriormente en la ACL del objeto de origen. Amazon S3 establece la ACL de la réplica de forma que se conceda control completo al propietario del bucket de destino.
- Al actualizar una configuración de replicación para habilitar o inhabilitar la invalidación del propietario, sucede lo siguiente.

  - Si añade la opción de invalidación del propietario a la configuración de replicación:

Cuando Amazon S3 replica una versión del objeto, descarta la ACL asociada al objeto de origen. En su lugar, establece la ACL de la réplica de forma que se conceda control completo al propietario del bucket de destino. No replica los cambios que se realicen posteriormente en la ACL del objeto de origen. No obstante, este cambio a la ACL no se aplica a las versiones de objetos que se replicaron antes de configurar la opción de invalidación del propietario. Las actualizaciones de las ACL de los objetos de origen que se replicaron antes de que se configurara la opción de invalidación del propietario se seguirán replicando (porque el objeto y sus réplicas siguen teniendo el mismo propietario).

- Si elimina la opción de invalidación del propietario de la configuración de replicación:

  Amazon S3 replica objetos nuevos que aparecen en el bucket de origen y las ACL asociadas a los buckets de destino. Para objetos que se replicaron antes de que eliminar la invalidación del propietario, Amazon S3 no replica las ACL porque el cambio de titularidad del objeto que realizó Amazon S3 permanece en vigor. Es decir, las ACL aplicadas a la versión del objeto que se replicaban cuando se estableció la opción de invalidación del propietario siguen sin replicarse.

Cómo cambiar al propietario de la réplica

Cuando los buckets de *`origen`* y *`destino`* de una configuración de replicación son propiedad de diferentes Cuentas de AWS, puede indicar a Amazon S3 que cambie la propiedad de la réplica a la Cuenta de AWS que posee el bucket de *`destino`*. En este ejemplo, se explica cómo utilizar la consola de Amazon S3 y la AWS CLI para cambiar la propiedad de la réplica. Para obtener más información, consulte Cambiar el propietario de la réplica.

> **Note**
>
> Cuando utiliza la replicación de S3 y los buckets de origen y destino pertenecen a diferentes Cuentas de AWS, el propietario del bucket de destino puede desactivar las ACL (con la configuración de propietario del bucket obligatorio para Object Ownership) para cambiar la propiedad de réplica a la Cuenta de AWS que posee el bucket de destino. Esta configuración imita el comportamiento de anulación del propietario existente sin necesidad del permiso `s3:ObjectOwnerOverrideToBucketOwner`. Esto significa que todos los objetos que se replican en el bucket de destino con la configuración de propietario del bucket obligatorio pertenecen al propietario del bucket de destino. Para obtener más información acerca de la propiedad de objetos, consulte Control de la propiedad de los objetos y desactivación de las ACL del bucket.

Para obtener más información acerca de la configuración de la replicación mediante el cifrado del lado del servidor con AWS Key Management Service en escenarios entre cuentas, consulte Conceder permisos adicionales para escenarios que afectan a varias cuentas.

Uso de la consola de S3

Para obtener instrucciones paso a paso, consulte Configuración de la replicación para buckets de origen y destino que son propiedad de la misma cuenta. En este tema, se proporcionan instrucciones para establecer la configuración de replicación cuando los buckets son propiedad de la misma y de diferentes Cuentas de AWS.

Uso de la AWS CLI

Para cambiar la propiedad de la réplica con la AWS CLI, es necesario crear buckets, habilitar el control de versiones en los buckets, crear un rol de IAM que conceda permiso a Amazon S3 para replicar objetos y agregar la configuración de replicación al bucket de origen. En la configuración de replicación indica a Amazon S3 que cambie la titularidad de la réplica. Ahora puede probar la configuración.

Parar cambiar la propiedad de la réplica cuando los buckets de origen y destino son propiedad de Cuentas de AWS (AWS CLI) diferentes

1. En este ejemplo, crea los buckets de *`origen`* y *`destino`* en dos Cuentas de AWS diferentes. Configure la AWS CLI con dos perfiles con nombre. Este ejemplo usa los nombres de perfil `acctA` y `acctB` respectivamente. Para obtener más información acerca de la configuración de perfiles de credenciales, consulte Perfiles con nombre en la Guía del usuario de AWS Command Line Interface.

   > **Important**
   >
   > Los perfiles utilizados para este ejercicio tienen que tener los permisos necesarios. Por ejemplo, en la configuración de replicación especifica el rol de IAM que Amazon S3 puede asumir. Solo puede hacer esto si el perfil que utiliza tiene el permiso `iam:PassRole`. Si utiliza credenciales de usuario de administrador para crear un perfil con nombre, puede realizar todas las tareas. Para obtener más información, consulte Concesión de permisos a un usuario para transferir un rol a un servicio de AWS en la Guía del usuario de IAM.

Tendrá que asegurarse de que estos permisos tengan los permisos necesarios. Por ejemplo, la configuración de replicación incluye un rol de IAM que Amazon S3 puede asumir. El perfil con nombre utilizado para asociar dicha configuración a un bucket solo puede hacerlo si tiene el permiso `iam:PassRole`. Si especifica credenciales de usuario de administrador al crear estos perfiles con nombre, entonces tienen todos los permisos. Para obtener más información, consulte Concesión de permisos a un usuario para transferir un rol a un servicio de AWS en la Guía del usuario de IAM.

2. Cree el bucket de *origen* y habilite el control de versiones. En este ejemplo, se crea el bucket de *origen* en la región EE. UU. Este (Norte de Virginia) (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. Cree un bucket de *destino* y habilite el control de versiones. En este ejemplo, se crea el bucket de *destino* en la región EE. UU. Oeste (Oregón) (us-west-2). Utilice un perfil de Cuenta de AWS diferente al utilizado para el bucket de *origen*.

```
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. Debe añadir permisos a la política del bucket de *destino* para permitir el cambio de titularidad de la réplica.

a. Guarde la siguiente política en *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. Coloque la política anterior en el bucket de *destino*:

```
aws s3api put-bucket-policy --region $ {destination_region} --
bucket $ {destination} --policy file://destination_bucket_policy.json
```

5. Cree un rol de IAM. Especifique este rol en la configuración de replicación que agregue al bucket de *origen* más adelante. Amazon S3 asume este rol para replicar objetos en su nombre. Crea el rol de IAM en dos pasos:

   - Crear un rol.
   - Asocie una política de permisos al rol.

   a. Cree un rol de IAM.

i. Copie la siguiente política de confianza y guárdela en un archivo llamado `s3-role-trust-policy.json` en el directorio actual en su equipo local. Esta política concede a Amazon S3 permisos para asumir el rol.

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

ii. Ejecute el siguiente comando de AWS CLI para crear un rol.

```
$ aws iam create-role \
--role-name replicationRole \
--assume-role-policy-document file://s3-role-trust-policy.json  \
--profile acctA
```

b. Asocie una política de permisos al rol.

i. Copie la siguiente política de permisos y guárdela en un archivo llamado `s3-role-perm-pol-changeowner.json` en el directorio actual en su equipo local. Esta política concede permisos para varias acciones de buckets y objetos de Amazon S3. En los siguientes pasos, crea un rol de IAM y asocia esta política al rol.

```
{
   "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. Para crear una política y asociarla al rol, ejecute el siguiente comando.

```
$ aws iam put-role-policy \
--role-name replicationRole \
--policy-document file://s3-role-perm-pol-changeowner.json \
--policy-name replicationRolechangeownerPolicy \
--profile acctA
```

6. Agregar una configuración de replicación al bucket de origen.

   a. La AWS CLI requiere que especifique la configuración de replicación como JSON. Guarde la siguiente JSON en un archivo denominado `replication.json` en el directorio actual en su equipo local. En la configuración, la adición de `AccessControlTranslation` para indicar un cambio en la titularidad de la réplica.

```
{
```

```
   "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. Edite la JSON proporcionando valores para el ID de la cuenta del propietario del bucket de *destino* y *ARN-rol-IAM*. Guarde los cambios.

c. Para añadir la configuración de replicación al bucket de origen, ejecute el siguiente comando. Proporcione el nombre del bucket de *origen*.

```
$ aws s3api put-bucket-replication \
--replication-configuration file://replication.json \
--bucket source \
--profile acctA
```

7. Compruebe la propiedad de la réplica en la consola de Amazon S3.

a. Inicie sesión AWS Management Console Management Console y abra la consola de Amazon S3 en https://console.aws.amazon.com/s3/.

b. Agregar objetos al bucket de *origen*. Compruebe que el bucket de *destino* contenga réplicas del objeto y que la propiedad de las réplicas haya cambiado a la Cuenta de AWS propietaria del bucket de *destino*.

Uso de los AWS SDK

Para informarse sobre un ejemplo de código que agregar a la configuración de replicación, consulte Uso de los AWS SDK. Tiene que modificar la configuración de replicación en concordancia. Para obtener información conceptual, consulte Cambiar el propietario de la réplica.

Cumplimiento de los requisitos de conformidad mediante el control de tiempo de replicación de S3 (S3 RTC)

S3 RTC (Control del tiempo de replicación de S3) le ayuda a cumplir requisitos de conformidad o requisitos empresariales para la replicación de datos y proporciona visibilidad de los tiempos de replicación de Amazon S3. S3 RTC replica la mayoría de los objetos que se cargan en Amazon S3 en unos segundos y el 99,99 % de esos objetos en un plazo de 15 minutos.

De forma predeterminada, S3 RTC incluye métricas de replicación de S3 y notificaciones de eventos de Amazon S3, que puede utilizar para supervisar la cantidad total de operaciones de la API de S3 con replicación pendiente, el tamaño total de los objetos con replicación pendiente, así como el tiempo máximo de replicación. Puede habilitar las métricas de replicación independientemente de S3 RTC. Para obtener más información, consulte Monitoreo del progreso con métricas de replicación. Además, S3 RTC proporciona eventos `OperationMissedThreshold` y `OperationReplicatedAfterThreshold` que notifican al propietario del bucket si la replicación de objetos supera o se replica después del umbral de 15 minutos.

Con S3 RTC, los eventos de Amazon S3 le pueden notificar los pocos casos en que los objetos no se replican en 15 minutos y cuando esos objetos se replican después del umbral de 15 minutos. Los eventos de Amazon S3 están disponibles a través de Amazon SQS, Amazon SNS o AWS Lambda. Para obtener más información, consulte the section called “Notificaciones de eventos de Amazon S3”.

Temas

- Control del tiempo de replicación de S3
- Métricas de replicación con S3 RTC
- Uso de notificaciones de eventos de Amazon S3 para realizar un seguimiento de objetos de replicación
- Prácticas recomendadas y directrices para S3 RTC
- Habilitación del control del tiempo de replicación de S3 (S3 RTC)

Control del tiempo de replicación de S3

Puede comenzar a utilizar S3 Replication Time Control (S3, RTC, Control del tiempo de replicación de S3) con una regla de replicación nueva o existente. Puede optar por aplicar la regla de replicación a un bucket de S3 completo o a objetos de Amazon S3 con un prefijo o etiqueta específicos. Cuando habilita S3 RTC, las métricas de replicación también se habilitan en la regla de replicación.

Si utiliza la última versión de la configuración de replicación (es decir, si especifica el `Filter` elemento en una regla de configuración de replicación), Amazon S3 no replicará el marcador de eliminación de forma predeterminada. Sin embargo, puede agregar la replicación de marcador de eliminación a reglas no basadas en etiquetas.

> **Note**
>
> Estas métricas de replicación se facturan al mismo precio que las métricas personalizadas de Amazon CloudWatch. Para obtener más información, consulte los precios de Amazon CloudWatch.

Para obtener más información acerca de cómo crear una regla con S3 RTC, consulte Habilitación del control del tiempo de replicación de S3 (S3 RTC).

Métricas de replicación con S3 RTC

Las reglas de replicación que tengan habilitado S3 Replication Time Control (S3 RTC, Control del tiempo de replicación de S3) pueden publicar métricas de replicación. Con las métricas de replicación, puede supervisar el número total de operaciones de la API de S3 que están pendientes de replicación, el tamaño total de los objetos pendientes de replicación, el tiempo máximo de replicación en la región de destino y el número total de operaciones que no se ha podido replicar. A continuación, puede monitorizar cada conjunto de datos que se replica por separado.

Las métricas de replicación están disponibles dentro de los 15 minutos siguientes a la habilitación de S3 RTC. Las métricas de replicación están disponibles a través de la consola de Amazon S3, la API de Amazon S3, los AWS SDK, la AWS Command Line Interface (AWS CLI) y Amazon CloudWatch. Para obtener más información, consulte Monitorización de métricas con Amazon CloudWatch.

Para obtener más información sobre cómo buscar métricas de replicación a través de la consola de Amazon S3, consulte Visualización de métricas de replicación mediante la consola de Amazon S3.

Uso de notificaciones de eventos de Amazon S3 para realizar un seguimiento de objetos de replicación

Puede realizar un seguimiento del tiempo de replicación de los objetos que no se replicaron en 15 minutos monitoreando las notificaciones de eventos específicas que S3 Replicacion Time Control (S3 RTC, Control del tiempo de replicación) publica. Estos eventos se publican cuando un objeto que cumplía los requisitos para la replicación mediante S3 RTC no se ha replicado en el plazo de 15 minutos y cuando ese objeto se replica después del umbral de 15 minutos.

Las métricas de replicación están disponibles dentro de los 15 minutos siguientes a la habilitación de S3 RTC. Los eventos de Amazon S3 están disponibles a través de Amazon SQS, Amazon SNS o AWS Lambda. Para obtener más información, consulte Notificaciones de eventos de Amazon S3.

Prácticas recomendadas y directrices para S3 RTC

Al replicar datos en Amazon S3 mediante S3 Replication Time Control (S3 RTC, Control del tiempo de replicación de S3), siga estas prácticas recomendadas para optimizar el rendimiento de replicación de sus cargas de trabajo.

Temas

Directrices para optimizar la tasa de solicitudes y el rendimiento de replicación de Amazon S3

Al cargar y recuperar almacenamiento de Amazon S3, sus aplicaciones pueden lograr fácilmente miles de transacciones por segundo en el rendimiento de la solicitud. Por ejemplo, una aplicación puede lograr al menos 3500 solicitudes PUT/COPY/POST/DELETE o 5500 GET/HEAD por segundo y prefijo en un bucket de S3, incluidas las solicitudes que realiza la replicación S3 en su nombre. No existe ningún límite en cuanto al número de prefijos dentro de un bucket. Puede aumentar el rendimiento de lectura o escritura ejecutando en paralelo las operaciones de lectura. Por ejemplo, si crea 10 prefijos en un bucket de S3 para ejecutar en paralelo las operaciones de lectura, podría escalar el rendimiento de lectura a 55.000 solicitudes de lectura por segundo.

Amazon S3 se escala automáticamente en respuesta a las tasas de solicitudes sostenidas por encima de estas directrices, o de tasas de solicitudes sostenidas simultáneas con solicitudes LIST.

Aunque Amazon S3 se está optimizando internamente para una nueva velocidad de solicitudes, podría recibir respuestas a las solicitudes HTTP 503 de forma temporal hasta que se complete la optimización. Esto puede ocurrir cuando se producen aumentos en las tasas de solicitudes por segundo o cuando se habilita S3 RTC por primera vez. Durante estos periodos, la latencia de replicación puede aumentar. El acuerdo de nivel de servicio (SLA) de S3 RTC no se aplica a los periodos en los que se exceden las directrices de rendimiento de Amazon S3 sobre solicitudes por segundo.

El SLA de S3 RTC tampoco se aplica durante los periodos de tiempo en los que la velocidad de transferencia de datos de replicación supera el límite predeterminado de 1 Gbps. Si prevé que la tasa de transferencia de replicación superará 1 Gbps, puede contactar con el Centro de AWS Support o utilizar Service Quotas para solicitar un aumento del límite.

Cálculo de las tasas de solicitudes de replicación

Su tasa total de solicitudes, incluidas las solicitudes que realiza la replicación de Amazon S3 en su nombre, debe estar comprendida en las directrices de tasa de solicitudes de Amazon S3 para los buckets de origen y destino de la replicación. Para cada objeto replicado, la replicación de Amazon S3 realiza hasta cinco solicitudes GET/HEAD y una solicitud PUT al bucket de origen y una solicitud PUT destinada a cada bucket de destino.

Por ejemplo, si prevé replicar 100 objetos por segundo, la replicación de Amazon S3 podría realizar 100 solicitudes PUT adicionales en su nombre para un total de 200 solicitudes PUT por segundo al bucket de S3 de origen. Además, la replicación de Amazon S3 puede realizar hasta 500 solicitudes GET/HEAD (5 solicitudes GET/HEAD por cada objeto replicado).

> **Note**
>
> Usted incurre en costos solamente por una solicitud PUT por cada objeto replicado. Para obtener más información, consulte la información de precios en las preguntas frecuentes sobre replicación de Amazon S3.

Superación de los límites de la tasa de transferencia de datos S3 RTC

Si prevé que la tasa de transferencia de datos de control de tiempo de replicación de S3 superará el límite predeterminado de 1 Gbps, póngase en contacto con el Centro deAWS Support o utilice Service Quotas para solicitar un aumento del límite.

Tasas de solicitudes de replicación de objetos cifrados con AWS KMS

Cuando replica objetos cifrados con cifrado del lado del servidor (SSE-KMS) mediante la replicación de Amazon S3, se aplican los límites de solicitudes por segundo de AWS Key Management Service (AWS KMS). AWS KMS podría rechazar una solicitud válida si su tasa de solicitudes excede el límite del número de solicitudes por segundo. Cuando se limita una solicitud de forma controlada, AWS KMS devuelve un error `ThrottlingException`. El límite de la tasa de solicitudes de AWS KMS se aplica a las solicitudes que usted realiza directamente y a aquellas que la replicación de Amazon S3 efectúa en su nombre.

Por ejemplo, si prevé replicar 1000 objetos por segundo, puede restar 2000 solicitudes del límite de tasa de solicitudes de AWS KMS. La tasa de solicitudes por segundo resultante estará disponible para las cargas de trabajo de AWS KMS, excluida la replicación. Puede utilizar las métricas de solicitudes de AWS KMS en Amazon CloudWatch para monitorear la tasa total de solicitudes de AWS KMS en la Cuenta de AWS.

Habilitación del control del tiempo de replicación de S3 (S3 RTC)

S3 RTC (Control del tiempo de replicación de S3) le ayuda a cumplir requisitos de conformidad o requisitos empresariales para la replicación de datos y proporciona visibilidad de los tiempos de replicación de Amazon S3. S3 RTC replica la mayoría de los objetos que se cargan en Amazon S3 en unos segundos y el 99,99 % de esos objetos en un plazo de 15 minutos.

Con S3 RTC, puede monitorizar el número total y el tamaño de los objetos que están pendientes de replicación, así como el tiempo máximo de replicación en la región de destino. Las métricas de replicación se encuentran disponibles en la AWS Management Console y la Guía del usuario de Amazon CloudWatch. Para obtener más información, consulte the section called “Métricas de replicación de S3 en CloudWatch ”.

Uso de la consola de S3

Para obtener instrucciones paso a paso, consulte Configuración de la replicación para buckets de origen y destino que son propiedad de la misma cuenta. En este tema, se proporcionan instrucciones para habilitar S3 RTC en la configuración de replicación cuando los buckets son propiedad de la misma y de diferentes Cuentas de AWS.

Uso de la AWS CLI

Para utilizar la AWS CLI con el propósito de replicar objetos con S3 RTC habilitado, es necesario crear buckets, habilitar el control de versiones en los buckets, crear un rol de IAM que conceda permiso a Amazon S3 para replicar objetos y agregar la configuración de replicación al bucket de

origen. La configuración de replicación debe tener habilitado el control de tiempo de replicación de S3 (S3 RTC).

Para replicar con S3 RTC habilitado (AWS CLI)

- En el ejemplo siguiente se establece `ReplicationTime` y `Metric`, luego se agrega la configuración de replicación al bucket de origen.

```
{
    "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**
>
> `Metrics:EventThreshold:Minutes` y `ReplicationTime:Time:Minutes` solo aceptan 15 como valor válido.

Uso de AWS SDK para Java

En el siguiente ejemplo de Java se agrega la configuración de replicación con el control del tiempo de replicación de 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);
  }
}
```

Para obtener más información, consulte Cumplimiento de los requisitos de conformidad mediante el control de tiempo de replicación de S3 (S3 RTC).

Replicación de objetos cifrados (SSE-C, SSE-S3, SSE-KMS, DSSE-KMS)

> **Important**
>
> Amazon S3 aplica ahora el cifrado del servidor con claves administradas por Amazon S3 (SSE-S3) como el nivel básico de cifrado para cada bucket de Amazon S3. Desde el 5 de enero de 2023, todas las cargas de objetos nuevos a Amazon S3 se cifran automáticamente sin costo adicional y sin afectar al rendimiento. El estado de cifrado automático para la configuración de cifrado predeterminada en el bucket de S3 y para cargas de objetos nuevos está disponible en registros de AWS CloudTrail, Inventario de S3, Lente de almacenamiento de S3, la consola de Amazon S3 y como encabezado de respuesta a la API de Amazon S3 adicional en AWS Command Line Interface y los SDK de AWS. Para obtener más información, consulte Preguntas frecuentes del cifrado predeterminado.

Existen algunas consideraciones especiales cuando replica objetos que se han cifrado mediante el cifrado en el servidor. Amazon S3 admite los siguientes tipos de cifrado en el servidor:

- Cifrado en el servidor con claves administradas por Amazon S3 (SSE-S3)
- Cifrado del lado del servidor con claves AWS Key Management Service (AWS KMS) (SSE-KMS)
- Cifrado del servidor de doble capa con claves de AWS KMS (DSSE-KMS)
- Cifrado en el servidor con claves proporcionadas por el cliente (SSE-C)

Para obtener más información acerca del cifrado del lado del servidor, consulte the section called “Cifrado en el servidor”.

En este tema se explican los permisos que necesita para dirigir Amazon S3 a la replicación de objetos que se han cifrado mediante el cifrado en el servidor. En este tema también se proporcionan

elementos de configuración adicionales que puede agregar y políticas de AWS Identity and Access Management (IAM) de ejemplo que conceden los permisos necesarios para replicar objetos cifrados.

Para ver un ejemplo con instrucciones paso a paso, consulte Habilitación de la replicación de objetos cifrados. Para obtener información acerca de la creación de configuración de replicación, consulte Información general de la replicación de objetos.

> **Note**
>
> Puede utilizar AWS KMS keys de varias regiones en Amazon S3. No obstante, Amazon S3 trata las claves de varias regiones como si fueran claves de una sola región y no utiliza las características de varias regiones de la clave. Para obtener más información, consulte Uso de claves de varias regiones en la Guía para desarrolladores de AWS Key Management Service.

Temas

- Cómo afecta el cifrado de buckets predeterminado a la replicación
- Replicación de objetos cifrados con SSE-C
- Replicación de objetos cifrados con SSE-S3, SSE-KMS o DSSE-KMS
- Habilitación de la replicación de objetos cifrados

Cómo afecta el cifrado de buckets predeterminado a la replicación

Cuando habilita el cifrado predeterminado para un bucket de destino de replicación, se aplica el siguiente comportamiento de cifrado:

- Si los objetos del bucket de origen no están cifrados, los objetos de réplica del bucket de destino se cifran mediante la configuración de cifrado predeterminado del bucket de destino. Como resultado, las etiquetas de entidad (ETags) de los objetos de origen difieren de las ETags de los objetos de réplica. Si tiene aplicaciones que utilizan ETags, deberá actualizarlas para tener en cuenta esta diferencia.
- Si los objetos del bucket de origen se cifran mediante el cifrado del servidor con claves administradas por Amazon S3 (SSE-S3), el cifrado del servidor con claves de AWS Key Management Service (AWS KMS) (SSE-KMS) o con cifrado del servidor de doble capa con claves de AWS KMS (DSSE-KMS), los objetos de réplica del bucket de destino utilizarán el mismo tipo

de cifrado que los objetos de origen. La configuración de cifrado predeterminado del bucket de destino no se utiliza.

Replicación de objetos cifrados con SSE-C

Al utilizar cifrado en el servidor con claves proporcionadas por el cliente (SSE-C), puede administrar sus propias claves de cifrado. Con SSE-C, usted administra las claves mientras que Amazon S3 administra el proceso de cifrado y descifrado. Debe proporcionar una clave de cifrado como parte de su solicitud, pero no necesita escribir ningún código para realizar el cifrado o descifrado de objetos. Cuando carga un objeto, Amazon S3 cifra el objeto mediante la clave que ha proporcionado. A continuación, Amazon S3 purga dicha clave de la memoria. Al recuperar un objeto, debe facilitar la misma clave de cifrado como parte de la solicitud. Para obtener más información, consulte the section called “Claves de cifrado proporcionadas por el cliente (SSE-C)”.

La replicación de S3 admite objetos cifrados con SSE-C. Puede configurar la replicación de objetos SSE-C en la consola de Amazon S3 o con los SDK de AWS, de la misma manera que configura la replicación para objetos no cifrados. No hay permisos de SSE-C adicionales a los que se requieren actualmente para la replicación.

La replicación de S3 replica automáticamente los objetos cifrados con SSE-C recién cargados si son elegibles, tal como se especifique en la configuración de replicación de S3. Para replicar objetos existentes en sus buckets, utilice la replicación por lotes de S3. Para obtener más información sobre la replicación de objetos, consulte the section called “Configuración de la replicación en directo” y the section called “Replicación de objetos existentes”.

No se aplican cargos adicionales por replicar objetos SSE-C. Para obtener más información sobre los precios de replicación, consulte la página de precios de Amazon S3.

Replicación de objetos cifrados con SSE-S3, SSE-KMS o DSSE-KMS

De forma predeterminada, Amazon S3 no replica objetos cifrados con SSE-KMS o DSSE-KMS. En esta sección se explican los elementos de configuración adicionales que puede agregar para indicar a Amazon S3 que replique estos objetos.

Para ver un ejemplo con instrucciones paso a paso, consulte Habilitación de la replicación de objetos cifrados. Para obtener información acerca de la creación de configuración de replicación, consulte Información general de la replicación de objetos.

Especificar información adicional en la configuración de replicación

En la configuración de replicación, haga lo siguiente:

- En el elemento `Destination` de su configuración de replicación, agregue el ID de la clave de AWS KMS simétrica administrada por el cliente que desea que Amazon S3 utilice para cifrar las réplicas de objetos, como se muestra en el siguiente ejemplo de configuración de replicación.
- Opte explícitamente por permitir la replicación de objetos cifrados mediante claves de KMS (SSE-KMS o DSSE-KMS). Para ello, agregue el elemento `SourceSelectionCriteria`, como se muestra en el siguiente ejemplo de configuración de la replicación.

```
<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
 Región de AWS as the destination bucket.</ReplicaKmsKeyID>
          </EncryptionConfiguration>
       </Destination>
      ...
   </Rule>
</ReplicationConfiguration>
```

Important

La clave de KMS debe haberse creado en la misma Región de AWS que el bucket de destino.

La clave de KMS debe ser válida. La operación de la API `PutBucketReplication` no comprueba la validez de las claves de KMS. Si usa una clave de KMS no válida, recibirá el código de estado `200 OK` de HTTP en respuesta, pero la replicación genera un error.

En el siguiente ejemplo se muestra una configuración de replicación que incluye elementos de configuración opcionales. Esta configuración de replicación tiene una regla. La regla se aplica a los objetos con el prefijo de clave `Tax`. Amazon S3 utiliza el ID de AWS KMS key especificado para cifrar estas réplicas de objetos.

```
<?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:::amzn-s3-demo-destination-bucket</Bucket>
         <EncryptionConfiguration>
            <ReplicaKmsKeyID>AWS KMS key ARN or Key Alias ARN that's in the same Región
 de 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>
```

Conceder permisos adicionales para el rol de IAM

Para replicar objetos cifrados en reposo mediante SSE-S3, SSE-KMS o DSSE-KMS, conceda los siguientes permisos adicionales al rol de AWS Identity and Access Management (IAM) que especifique en la configuración de replicación. Estos permisos los otorga actualizando la política de permisos asociada con el rol de IAM.

- Acción **`s3:GetObjectVersionForReplication`** para objetos de origen: esta acción permite que Amazon S3 replique tanto los objetos sin cifrar como los creados con cifrado del servidor mediante claves de SSE-S3, SSE-KMS o DSSE-KMS.

  > **Note**
  >
  > Se recomienda utilizar la `s3:GetObjectVersionForReplication` acción en lugar de la acción `s3:GetObjectVersion`, ya que `s3:GetObjectVersionForReplication` proporciona a Amazon S3 solo los permisos mínimos necesarios para la replicación. Además, la acción `s3:GetObjectVersion` permite replicar objetos sin cifrar y cifrados con SSE-S3, pero no replicar objetos cifrados con claves de KMS (SSE-KMS o DSSE-KMS).

- Acciones **`kms:Decrypt`** y **`kms:Encrypt`** de AWS KMS para las claves de KMS
  - Debe conceder permisos `kms:Decrypt` para la AWS KMS key que se utilizó para descifrar el objeto de origen.
  - Debe conceder permisos `kms:Encrypt` para la AWS KMS key que se utilizó para cifrar la réplica del objeto.
- Acción **`kms:GenerateDataKey`** para replicar objetos de texto sin formato: si está replicando objetos de texto sin formato en un bucket con el cifrado SSE-KMS o DSSE-KMS habilitado de forma predeterminada, debe incluir el permiso `kms:GenerateDataKey` para el contexto de cifrado de destino y la clave de KMS en la política de IAM.

Se recomienda restringir estos permisos solo a los buckets y objetos de destino que utilicen las claves de condición de AWS KMS. La Cuenta de AWS que posea el rol de IAM debe tener permisos para la acciones `kms:Encrypt` y `kms:Decrypt` para las claves de KMS que se indican en la política. Si las claves de KMS pertenecen a otra cuenta de Cuenta de AWS, el propietario de las claves de KMS debe conceder estos permisos a la Cuenta de AWS que posee el rol de IAM. Para obtener más información acerca de cómo administrar el acceso a estas claves de KMS, consulte Uso de políticas de IAM con AWS KMS en la Guía para desarrolladores de AWS Key Management Service.

Claves de bucket y replicación de S3

Para utilizar la replicación con una clave de Bucket de S3, la política de AWS KMS key para la clave de KMS utilizada para cifrar la réplica de objeto debe incluir el permiso `kms:Decrypt` para la entidad principal que realiza la llamada. La llamada a `kms:Decrypt` verifica la integridad de la clave de

bucket de S3 antes de usarla. Para obtener más información, consulte Uso de una clave de bucket de S3 con replicación.

Cuando se habilita una clave de bucket de S3 para el bucket de origen o de destino, el contexto de cifrado será el nombre de recurso de Amazon (ARN) del bucket y no el ARN del objeto (por ejemplo, `arn:aws:s3:::bucket_ARN`). Debe actualizar las políticas de IAM para usar el ARN del bucket para el contexto de cifrado:

```
"kms:EncryptionContext:aws:s3:arn": [
"arn:aws:s3:::bucket_ARN"
]
```

Para obtener más información, consulte Contexto de cifrado (`x-amz-server-side-encryption-context`) (en la sección “Uso de la API de REST”) y Cambios para tener en cuenta antes de habilitar una clave de bucket de S3.

Ejemplo de políticas: uso de SSE-KMS y SSE-KMS con replicación

En las siguientes políticas de IAM de ejemplo se muestran instrucciones para utilizar SSE-S3 y SSE-KMS con replicación.

Example : uso de SSE-KMS con buckets de destino independientes

En la siguiente política de ejemplo, se muestran instrucciones para utilizar SSE-KMS con buckets de destino independientes.

```
{
   "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:::amzn-s3-demo-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:::amzn-s3-demo-destination-bucket1/key-prefix1*"
                    ]
            }
            },
            "Resource": [
                "AWS KMS key ARNs (in the same Región de 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:::amzn-s3-demo-destination-bucket2/key-prefix1*"
                    ]
            }
            },
            "Resource": [
                "AWS KMS key ARNs (in the same Región de AWS as destination bucket 2). Used
 to encrypt object replicas created in destination bucket 2."
            ]
        }
    ]
}
```

Example : replicación de objetos creados con SSE-S3 y SSE-KMS

A continuación, se muestra una política de IAM completa que concede los permisos necesarios para replicar objetos no cifrados, objetos creados con SSE-KMS y objetos creados con SSE-KMS.

```
{
```

```
   "Version":"2012-10-17",
   "Statement":[
      {
         "Effect":"Allow",
         "Action":[
            "s3:GetReplicationConfiguration",
            "s3:ListBucket"
         ],
         "Resource":[
            "arn:aws:s3:::amzn-s3-demo-source-bucket"
         ]
      },
      {
         "Effect":"Allow",
         "Action":[
            "s3:GetObjectVersionForReplication",
            "s3:GetObjectVersionAcl"
         ],
         "Resource":[
            "arn:aws:s3:::amzn-s3-demo-source-bucket/key-prefix1*"
         ]
      },
      {
         "Effect":"Allow",
         "Action":[
            "s3:ReplicateObject",
            "s3:ReplicateDelete"
         ],
         "Resource":"arn:aws:s3:::amzn-s3-demo-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:::amzn-s3-demo-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:::amzn-s3-demo-destination-bucket/prefix1*"
               ]
            }
         },
         "Resource":[
            "AWS KMS key ARNs (in the same Región de AWS as the destination bucket) to
 use for encrypting object replicas"
         ]
      }
   ]
}
```

Example : replicación de objetos con claves de bucket de S3

A continuación, se muestra una política de IAM completa que concede los permisos necesarios para replicar objetos con claves de bucket de S3.

```
{
   "Version":"2012-10-17",
   "Statement":[
      {
         "Effect":"Allow",
         "Action":[
            "s3:GetReplicationConfiguration",
            "s3:ListBucket"
         ],
         "Resource":[
            "arn:aws:s3:::amzn-s3-demo-source-bucket"
         ]
      },
      {
```

```
         "Effect":"Allow",
         "Action":[
            "s3:GetObjectVersionForReplication",
            "s3:GetObjectVersionAcl"
         ],
         "Resource":[
            "arn:aws:s3:::amzn-s3-demo-source-bucket/key-prefix1*"
         ]
      },
      {
         "Effect":"Allow",
         "Action":[
            "s3:ReplicateObject",
            "s3:ReplicateDelete"
         ],
         "Resource":"arn:aws:s3:::amzn-s3-demo-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:::amzn-s3-demo-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:::amzn-s3-demo-destination-bucket"
```

```
                ]
             }
          },
          "Resource":[
             "AWS KMS key ARNs (in the same Región de AWS as the destination bucket) to
 use for encrypting object replicas"
          ]
       }
    ]
}
```

Conceder permisos adicionales para escenarios que afectan a varias cuentas

En un escenario de reproducción entre cuentas en el que los buckets de origen y destino pertenecen a Cuentas de AWS diferentes, puede utilizar una clave de KMS para cifrar réplicas de objetos. El propietario de la clave de KMS debe conceder al propietario del bucket de origen permiso para usar la clave de KMS.

> **Note**
>
> Si necesita replicar datos de SSE-KMS entre cuentas, la regla de replicación debe especificar una clave administrada por el cliente de AWS KMS para la cuenta de destino. Las Claves administradas por AWS no permiten el uso entre cuentas y, por tanto, no se pueden usar para realizar la replicación entre cuentas.

Para conceder permiso al propietario del bucket de origen para usar la clave de KMS (consola de AWS KMS)

1. Inicie sesión en la AWS Management Console y abra la consola de AWS KMS en https://console.aws.amazon.com/kms.
2. Para cambiar la Región de AWS, utilice el Selector de regiones ubicado en la esquina superior derecha de la página.
3. Si desea ver las claves de la cuenta que usted crea y administra, en el panel de navegación, elija Customer managed keys (Claves administradas por el cliente).
4. Seleccione la clave de KMS;.
5. En Configuración general, elija la pestaña Política de claves.
6. Desplácese hacia abajo hasta Otras Cuentas de AWS.

7. Elija Agregar otras Cuentas de AWS.

   Aparecerá el cuadro de diálogo Otro Cuentas de AWS.
8. En el cuadro de diálogo, elija Agregar otro Cuenta de AWS. Para arn:aws:iam::, introduzca el ID de la cuenta de bucket de origen.
9. Elija Guardar cambios.

Para conceder permiso al propietario del bucket de origen para usar la clave de KMS (AWS CLI)

- Para obtener información sobre el comando `put-key-policy` AWS Command Line Interface (AWS CLI), consulte put-key-policy en la Referencia de comandos de AWS CLI. Para obtener información acerca de la operación de la API `PutKeyPolicy` subyacente, consulte PutKeyPolicy en la Referencia de la API de AWS Key Management Service.

Consideraciones sobre cuotas de transacciones de AWS KMS

Cuando agregue muchos objetos nuevos con cifrado de AWS KMS después de activar la replicación entre regiones (CRR), es posible que experimente una limitación (errores `503 Service Unavailable` de HTTP). La limitación controlada se produce cuando el número de transacciones de AWS KMS por segundo supera las cuotas actuales. Para obtener más información, consulte Cuotas en la Guía para desarrolladores de AWS Key Management Service.

Para solicitar un aumento de una cuota, use Service Quotas. Para obtener más información, consulte Solicitud de un aumento de cuota. Si Service Quotas no es compatible en su región, abra un caso de AWS Support.

Habilitación de la replicación de objetos cifrados

De forma predeterminada, Amazon S3 no replica los objetos cifrados mediante el cifrado del servidor con claves de AWS Key Management Service (AWS KMS) (SSE-KMS) ni el cifrado del servidor de doble capa con claves de AWS KMS (DSSE-KMS). Para replicar objetos cifrados con SSE-KMS o DSS-KMS, es necesario modificar la configuración de replicación del bucket para indicar a Amazon S3 que replique estos objetos. Este ejemplo explica cómo usar la consola de Amazon S3 y la AWS Command Line Interface (AWS CLI) para cambiar la configuración de replicación del bucket con el fin de habilitar la replicación de objetos cifrados.

Para obtener más información, consulte Replicación de objetos cifrados (SSE-C, SSE-S3, SSE-KMS, DSSE-KMS).

> **Note**
>
> Cuando se habilita una clave de bucket de S3 para el bucket de origen o de destino, el contexto de cifrado será el nombre de recurso de Amazon (ARN) del bucket y no el ARN del objeto. Debe actualizar las políticas de IAM para usar el ARN del bucket para el contexto de cifrado. Para obtener más información, consulte Claves de bucket y replicación de S3.

> **Note**
>
> Puede utilizar AWS KMS keys de varias regiones en Amazon S3. No obstante, Amazon S3 trata las claves de varias regiones como si fueran claves de una sola región y no utiliza las características de varias regiones de la clave. Para obtener más información, consulte Uso de claves de varias regiones en la Guía para desarrolladores de AWS Key Management Service.

Uso de la consola de S3

Para obtener instrucciones paso a paso, consulte Configuración de la replicación para buckets de origen y destino que son propiedad de la misma cuenta. En este tema, se proporcionan instrucciones para establecer la configuración de replicación cuando los buckets son propiedad de la misma y de diferentes Cuentas de AWS.

Uso de la AWS CLI

Para replicar los objetos replicados cifrados con AWS CLI, haga lo siguiente:

- Cree los buckets de origen y de destino y habilite el control de versiones de dichos buckets.
- Cree un rol de servicio de AWS Identity and Access Management (IAM) que conceda permiso para replicar objetos en Amazon S3. Los permisos del rol de IAM incluye los permisos necesarios para replicar los objetos cifrados.
- Añada una configuración de replicación al bucket de origen. La configuración de replicación proporciona información relacionada con objetos de replicación cifrados mediante claves de KMS.
- Agregue objetos al bucket de origen.
- Pruebe la configuración para confirmar que los objetos cifrados se están replicando en el bucket de destino.

Los siguientes procedimientos le guiarán por este proceso.

Para replicar objetos cifrados del lado del servidor (AWS CLI)

1. En este ejemplo, se crea tanto el bucket *amzn-s3-demo-source-bucket* como el *amzn-s3-demo-destination-bucket* en la misma Cuenta de AWS. También puede configurar un perfil de credenciales para la AWS CLI. En este ejemplo, usamos el nombre de perfil *acctA*.

   Para obtener más información acerca de la configuración de perfiles de credenciales, consulte Perfiles con nombre en la Guía del usuario de AWS Command Line Interface. Para usar los comandos de este ejemplo, sustituya *user input placeholders* por su información.

2. Use los siguientes comandos para crear el bucket *DOC-EXAMPLE-SOURCE-BUCKET* y habilitar el control de versiones en él. Los siguientes comandos de ejemplo crean el bucket *DOC-EXAMPLE-SOURCE-BUCKET* en la región Este de EE. UU. (Norte de Virginia) (`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. Use los siguientes comandos para crear el bucket *DOC-EXAMPLE-DESTINATION-BUCKET* y habilitar el control de versiones en él. Los siguientes comandos de ejemplo crean el bucket *DOC-EXAMPLE-DESTINATION-BUCKET* en la región Oeste de EE. UU. (Oregón) (`us-west-2`).

   > **Note**
   >
   > Para establecer la configuración de replicación cuando los buckets *DOC-EXAMPLE-SOURCE-BUCKET* y *DOC-EXAMPLE-DESTINATION-BUCKET* están en la misma Cuenta de AWS, debe utilizar el mismo perfil. En este ejemplo, usaremos *acctA*. Para configurar la replicación cuando los buckets son propiedad de diferentes Cuentas de AWS, debe especificar diferentes perfiles para cada uno.

```
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. A continuación, cree un rol de servicio de IAM. Especificará este rol en la configuración de replicación que agregue al bucket *DOC-EXAMPLE-SOURCE-BUCKET* más adelante. Amazon S3 asume este rol para replicar objetos en su nombre. Crea el rol de IAM en dos pasos:

   - Cree un rol de servicio.
   - Asocie una política de permisos al rol.

   a. Para crear un rol de servicio de IAM, haga lo siguiente:

      i. Copie la siguiente política de confianza y guárdela en un archivo llamado `s3-role-trust-policy-kmsobj.json` en el directorio actual en su equipo local. Esta política concede permisos a la entidad principal de servicio de Amazon S3 para asumir el rol para que Amazon S3 puede realizar tareas en su nombre.

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

      ii. Utilice el siguiente comando para crear el rol:

```
$ aws iam create-role \
--role-name replicationRolekmsobj \
--assume-role-policy-document file://s3-role-trust-policy-kmsobj.json  \
--profile acctA
```

b. A continuación, asocie una política de permisos al rol. Esta política concede permisos para varias acciones de buckets y objetos de Amazon S3.

i. Copie la siguiente política de permisos y guárdela en un archivo llamado `s3-role-permissions-policykmsobj.json` en el directorio actual en su equipo local. Creará un rol de IAM y le asociará la política más adelante.

> **Important**
>
> En la política de permisos, debe especificar los ID de la clave AWS KMS que se emplearán para el cifrado de los buckets *amzn-s3-demo-source-bucket* y *amzn-s3-demo-destination-bucket*. Debe crear dos claves de KMS para los buckets *amzn-s3-demo-source-bucket* y *amzn-s3-demo-destination-bucket*. Las AWS KMS keys no se comparten fuera de la Región de AWS en la que se crearon.

```
{
   "Version":"2012-10-17",
   "Statement":[
      {
         "Action":[
            "s3:ListBucket",
            "s3:GetReplicationConfiguration",
            "s3:GetObjectVersionForReplication",
            "s3:GetObjectVersionAcl",
            "s3:GetObjectVersionTagging"
         ],
         "Effect":"Allow",
         "Resource":[
            "arn:aws:s3:::amzn-s3-demo-source-bucket",
            "arn:aws:s3:::amzn-s3-demo-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:::amzn-s3-demo-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:::amzn-s3-demo-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:::amzn-s3-demo-destination-bucket/*"
               ]
            }
         },
         "Resource":[
            "AWS KMS key IDs(in ARN format) to use for encrypting object
 replicas"
         ]
      }
   ]
}
```

ii. Cree una política y asóciela al rol.

```
$ aws iam put-role-policy \
--role-name replicationRolekmsobj \
--policy-document file://s3-role-permissions-policykmsobj.json \
--policy-name replicationRolechangeownerPolicy \
--profile acctA
```

5. A continuación, añada la siguiente configuración de replicación al bucket *amzn-s3-demo-source-bucket*, que le indica a Amazon S3 que replique los objetos con el prefijo `Tax/` en el bucket *amzn-s3-demo-destination-bucket*.

> **Important**
>
> En la configuración de replicación, debe especificar el rol de IAM que puede asumir Amazon S3. Solo puede hacer esto si tiene el permiso `iam:PassRole`. El perfil que especifique en el comando de la CLI tiene que tener este permiso. Para obtener más información, consulte Concesión de permisos a un usuario para transferir un rol a un servicio de Servicio de AWS en la Guía del usuario de 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:::amzn-s3-demo-destination-bucket</Bucket>
      <EncryptionConfiguration>
        <ReplicaKmsKeyID>AWS KMS key IDs to use for encrypting object replicas</
ReplicaKmsKeyID>
      </EncryptionConfiguration>
    </Destination>
  </Rule>
</ReplicationConfiguration>
```

Para añadir una configuración de replicación al bucket *amzn-s3-demo-source-bucket*, haga lo siguiente:

a. La AWS CLI requiere que especifique la configuración de replicación como JSON. Guarde la siguiente JSON en un archivo (`replication.json`) en el directorio actual en su equipo local.

```
{
   "Role":"IAM-Role-ARN",
   "Rules":[
      {
         "Status":"Enabled",
         "Priority":1,
         "DeleteMarkerReplication":{
            "Status":"Disabled"
         },
         "Filter":{
            "Prefix":"Tax"
         },
         "Destination":{
```

```
            "Bucket":"arn:aws:s3:::amzn-s3-demo-destination-bucket",
            "EncryptionConfiguration":{
               "ReplicaKmsKeyID":"AWS KMS key IDs (in ARN format) to use for
 encrypting object replicas"
            }
         },
         "SourceSelectionCriteria":{
            "SseKmsEncryptedObjects":{
               "Status":"Enabled"
            }
         }
      }
   ]
}
```

b. Edite el JSON para proporcionar valores para el bucket *amzn-s3-demo-destination-bucket*, *AWS KMS key IDs (in ARN format)* y *IAM-role-ARN*. Guarde los cambios.

c. Use el siguiente comando para añadir la configuración de replicación al bucket *amzn-s3-demo-source-bucket*. Asegúrese de proporcionar el nombre del bucket *amzn-s3-demo-source-bucket*.

```
$ aws s3api put-bucket-replication \
--replication-configuration file://replication.json \
--bucket amzn-s3-demo-source-bucket \
--profile acctA
```

6. Compruebe la configuración para verificar que se hayan replicado los objetos cifrados. En la consola de Amazon S3, haga lo siguiente:

a. Inicie sesión AWS Management Console Management Console y abra la consola de Amazon S3 en https://console.aws.amazon.com/s3/.

b. En el bucket *amzn-s3-demo-source-bucket*, cree una carpeta llamada `Tax`.

c. Añada objetos de ejemplo a la carpeta. Asegúrese de elegir la opción de cifrado y especifique su clave de KMS para cifrar los objetos.

d. Compruebe que el bucket *amzn-s3-demo-destination-bucket* contenga las réplicas de objeto y que se hayan cifrado con la clave de KMS que especificó en la configuración. Para obtener más información, consulte the section called “Obtener estado de replicación”.

Uso de los AWS SDK

Para ver un ejemplo de código sobre cómo agregar una configuración de replicación, consulte Uso de los AWS SDK. Tendrá que modificar la configuración de replicación en concordancia.

Para obtener información conceptual, consulte Replicación de objetos cifrados (SSE-C, SSE-S3, SSE-KMS, DSSE-KMS).

Replicación de cambios de metadatos con la sincronización de modificación de réplica de Amazon S3

La sincronización de modificación de réplicas de Amazon S3 puede ayudarle a mantener los metadatos de objetos, como etiquetas, ACL y configuración de bloqueo de objetos replicados entre réplicas y objetos de origen. De forma predeterminada, Amazon S3 replica metadatos de los objetos de origen únicamente a las réplicas. Cuando se habilita la sincronización de modificación de réplica, Amazon S3 replica los cambios de metadatos realizados en las copias de réplica en el objeto de origen, lo que hace que la replicación sea bidireccional.

Habilitación de la sincronización de modificación de réplica

Puede utilizar la sincronización de modificación de réplica de Amazon S3 con reglas de replicación nuevas o existentes. Puede aplicarlo a un bucket de S3 completo o a objetos de Amazon S3 que tengan un prefijo específico.

Para habilitar la sincronización de modificación de réplicas mediante la consola de Amazon S3, consulte Ejemplos para configurar la replicación en directo. En este tema, se proporcionan instrucciones para habilitar la sincronización de modificación de réplicas en la configuración de replicación cuando los buckets pertenecen a la misma o diferentes Cuentas de AWS.

Para habilitar la sincronización de modificación de réplicas mediante AWS Command Line Interface (AWS CLI), debe agregar una configuración de replicación al bucket que contenga las réplicas con `ReplicaModifications` habilitado. Para configurar la replicación bidireccional, cree una regla de replicación desde el bucket de origen (*`amzn-s3-demo-bucket1`*) hasta el bucket que contiene las réplicas (*`amzn-s3-demo-bucket2`*). A continuación, cree una segunda regla de replicación desde el bucket que contiene las réplicas (*`amzn-s3-demo-bucket2`*) hasta el bucket de origen (*`amzn-s3-demo-bucket1`*). Los buckets pueden estar en las mismas o en diferentes Regiones de AWS.

> **Note**
>
> Debe habilitar la sincronización de modificaciones de réplicas en ambos buckets para replicar los cambios en los metadatos de la réplica, como listas de control de acceso (ACL) a objetos,

etiquetas de objetos o configuraciones de bloqueo de objetos en los objetos replicados. Como todas las reglas de replicación, estas reglas se pueden aplicar a todo el bucket de Amazon S3 o a un subconjunto de objetos de Amazon S3 filtrados por prefijos o etiquetas de objeto.

En la siguiente configuración de ejemplo, Amazon S3 replica los cambios de metadatos con el prefijo *Tax* en el bucket *amzn-s3-demo-bucket*, que contendrá los objetos de origen.

```
{
    "Rules": [
        {
            "Status": "Enabled",
            "Filter": {
                "Prefix": "Tax"
            },
            "SourceSelectionCriteria": {
                "ReplicaModifications":{
                    "Status": "Enabled"
                }
            },
            "Destination": {
                "Bucket": "arn:aws:s3:::DOC-EXAMPLE-BUCKET"
            },
            "Priority": 1
        }
    ],
    "Role": "IAM-Role-ARN"
}
```

Para obtener instrucciones completas sobre la creación de reglas de replicación mediante la AWS CLI, consulte Configuración de la replicación para buckets de origen y destino que son propiedad de la misma cuenta.

Replicación de marcadores de eliminación entre buckets

De forma predeterminada, cuando se habilita la replicación de S3 y se elimina un objeto en el bucket de origen, Amazon S3 agrega un marcador de eliminación solo en el bucket de origen. Esta acción protege los datos de eliminaciones malintencionadas.

Si tiene habilitada la replicación de marcadores de eliminación, estos marcadores se copian en los buckets de destino y Amazon S3 se comporta como si el objeto se eliminara tanto en los buckets de origen como de destino. Para obtener más información sobre cómo funcionan los marcadores de eliminación, consulte Trabajar con marcadores de eliminación.

> **Note**
>
> La replicación de marcador de eliminación no es compatible con las reglas de replicación basadas en etiquetas. La replicación de marcador de eliminación no adhiere con el SLA de 15 minutos concedido al utilizar el control de tiempo de replicación de S3.

Si no utiliza la última versión de configuración de replicación, las operaciones de eliminación afectarán a la replicación de manera diferente. Para obtener más información, consulte Cómo afectan las operaciones de eliminación a la replicación.

Habilitar la replicación de marcador de eliminación

Puede comenzar a utilizar la replicación de marcador de eliminación con una regla de replicación nueva o existente. Puede aplicarlo a un bucket de S3 completo o a objetos de Amazon S3 que tengan un prefijo específico.

> **Note**
>
> Si habilita la replicación de marcadores de eliminación y el bucket tiene una regla de caducidad del ciclo de vida de S3, los marcadores de eliminación añadidos por esta regla no se replicarán en el bucket de destino.

Para habilitar la replicación de marcadores de eliminación mediante la consola de Amazon S3, consulte Uso de la consola de S3. En este tema, se proporcionan instrucciones para habilitar la replicación de marcador de eliminación en la configuración de replicación cuando los buckets pertenecen a la misma o a diferentes Cuentas de AWS.

Para habilitar la replicación de marcador de eliminación mediante la AWS Command Line Interface (AWS CLI), debe agregar una configuración de replicación al bucket de origen con `DeleteMarkerReplication` habilitado.

En la siguiente configuración de ejemplo, los marcadores de eliminación se replican en el bucket de destino *DOC-EXAMPLE-BUCKET* para los objetos con el prefijo *Tax*.

```
{
    "Rules": [
        {
            "Status": "Enabled",
            "Filter": {
                "Prefix": "Tax"
            },
            "DeleteMarkerReplication": {
                "Status": "Enabled"
            },
            "Destination": {
                "Bucket": "arn:aws:s3:::DOC-EXAMPLE-BUCKET"
            },
            "Priority": 1
        }
    ],
    "Role": "IAM-Role-ARN"
}
```

Para obtener instrucciones completas sobre la creación de reglas de replicación a través de la AWS CLI, consulte Configuración de la replicación para buckets de origen y destino que son propiedad de la misma cuenta en la sección de explicaciones de la replicación.

Administración o pausa de la replicación en directo

La replicación en directo consiste en la copia automática y asíncrona de los objetos de los buckets en las mismas o en diferentes Regiones de AWS. Tras configurar la configuración de replicación, Amazon S3 replica los objetos creados recientemente y las actualizaciones de objetos de un bucket de origen en uno o varios buckets de destino especificados.

Para agregar reglas de replicación al bucket de origen se utiliza la consola de Amazon S3. Las reglas de replicación definen los objetos del bucket de origen que se deben replicar y el bucket o buckets de destino donde se almacenan los objetos replicados. Para obtener más información acerca de la replicación, consulte Información general de la replicación de objetos.

Las reglas de replicación se administran en la página Replication (Replicación). Puede añadir, ver, habilitar, deshabilitar o eliminar las reglas de replicación. También puede cambiar la prioridad de las reglas de replicación. Para obtener información acerca de cómo agregar reglas de replicación a un bucket, consulte Uso de la consola de S3.

Administración de las reglas de replicación para un bucket de S3 mediante la consola de Amazon S3

1. Inicie sesión AWS Management Console Management Console y abra la consola de Amazon S3 en https://console.aws.amazon.com/s3/.
2. En el panel de navegación izquierdo, elija Instancias.
3. En la pestaña Buckets de uso general, elija el nombre del bucket que desee.
4. Elija la pestaña Administración y desplácese hacia abajo hasta Reglas de replicación.
5. Puede cambiar las reglas de replicación de las siguientes formas:
   - Para habilitar o deshabilitar una regla de replicación, elija el botón de opción a la izquierda de la regla. En el menú Acciones, seleccione Habilitar regla o Deshabilitar regla. También puede deshabilitar, habilitar o eliminar todas las reglas del bucket en el menú Acciones.

     > **Note**
     >
     > Si deshabilita una regla de replicación y, posteriormente, la vuelve a habilitar, los objetos nuevos o modificados que no se hayan replicado mientras la regla estaba deshabilitada no se replicarán automáticamente cuando se vuelva a habilitar la regla. Para replicar esos objetos, debe utilizar la replicación por lotes de S3. Para obtener más información, consulte the section called “Replicación de objetos existentes”.

   - Para cambiar la prioridad de una regla, pulse el botón de opción situado a la izquierda de la regla y, a continuación, elija Editar regla.

     Establece las prioridades de las reglas para evitar conflictos causados por objetos incluidos en el ámbito de más de una regla. En caso de solaparse las reglas, Amazon S3 usará la prioridad de la regla para determinar la regla que se va a aplicar. Cuanto mayor sea el número, mayor será la prioridad. Para obtener más información acerca de la prioridad de regla, consulte Configuración de replicación.

Pausa o detención de una replicación

Para pausar temporalmente una replicación y hacer que se reanude automáticamente más adelante, puede utilizar la acción de `aws:s3:bucket-pause-replication` en AWS Fault Injection Service. Para obtener más información, consulte aws:s3:bucket-pause-replication y Pausa de la replicación de S3 en la Guía del usuario de AWS Fault Injection Service.

Para detener la replicación en Amazon S3, le recomendamos que deshabilite las reglas de replicación. Si deshabilita una regla de replicación y, posteriormente, la vuelve a habilitar, los objetos nuevos o modificados que no se hayan replicado mientras la regla estaba deshabilitada no se replicarán automáticamente cuando se vuelva a habilitar la regla. Para replicar esos objetos, debe utilizar la replicación por lotes de S3. Para obtener más información, consulte the section called “Replicación de objetos existentes”.

La replicación también se detendrá si elimina el rol de AWS Identity and Access Management (IAM), los permisos de AWS Key Management Service (AWS KMS) o los permisos de la política de bucket que conceden a Amazon S3 los permisos necesarios. Sin embargo, no recomendamos estos métodos porque provocan errores en la replicación. Amazon S3 informa del estado de replicación de los objetos afectados como `FAILED`. Si los permisos se restauran posteriormente, los objetos marcados como `FAILED` no se replicarán automáticamente. Para replicar esos objetos, debe utilizar la replicación por lotes de S3.

Monitoreo del progreso con métricas de replicación y notificaciones de eventos de S3

Las métricas de replicación de S3 proporcionan métricas detalladas para las reglas de replicación en la configuración de la misma. Con las métricas de replicación, puede monitorizar su progreso minuto a minuto mediante el seguimiento de los bytes pendientes, las operaciones que no se han replicado y la latencia de replicación.

Las métricas de replicación de S3 se activan automáticamente cuando se habilita el control de tiempo de replicación de S3 (S3 RTC). También puede habilitar las métricas de replicación de S3 independientemente del RTC de S3 al crear o editar una regla. S3 RTC incluye otras características, como un acuerdo de nivel de servicio (SLA) y notificaciones de umbrales perdidos. Para obtener más información, consulte Cumplimiento de los requisitos de conformidad mediante el control de tiempo de replicación de S3 (S3 RTC).

Las métricas de bytes pendientes, de operaciones pendientes y de latencia de replicación se aplican únicamente a los objetos nuevos que se replican con la replicación entre regiones de S3 (S3 CRR) o la replicación en la misma región de S3 (S3 SRR). La métrica de replicación fallida de las operaciones rastrea tanto los objetos nuevos que se replican con S3 CRR o S3 SRR como los objetos existentes que se replican con la replicación por lotes de S3. También puede configurar notificaciones de eventos de Amazon S3 a fin de recibir eventos de error de replicación para ayudar a solucionar cualquier problema de configuración.

Cuando está habilitada, las métricas de replicación de S3 publican las siguientes métricas en Amazon CloudWatch:

- Bytes pendientes de replicación: número total de bytes de objetos pendientes de replicación para una regla de replicación determinada.
- Latencia de replicación: número máximo de segundos durante los cuales los buckets de destino de replicación están detrás del bucket de origen para una regla de replicación determinada.
- Operaciones pendientes de replicación: número de operaciones pendientes de replicación para una regla de replicación determinada. Esta métrica realiza un seguimiento de las operaciones relacionadas con los objetos, los marcadores de eliminación, las etiquetas, las listas de control de acceso (ACL) y el bloqueo de objetos de S3.
- Operaciones en las que se produjo un error en la replicación: número de operaciones en las que se ha producido un error de replicación para una regla de replicación determinada. Esta métrica realiza un seguimiento de las operaciones relacionadas con los objetos, los marcadores de eliminación, las etiquetas, las ACL y el bloqueo de objetos. A diferencia de las otras métricas de replicación, esta métrica aplica tanto los objetos nuevos que se replican con S3 CRR o S3 SRR como los objetos existentes que se replican con la replicación por lotes de S3.

> **Note**
>
> Operaciones en las que se produjo un error en la replicación rastrea los errores de replicación de S3 agregados en un intervalo por minuto. Para identificar los objetos específicos que han fallado la replicación y los motivos de dichos fallos, suscríbase al evento `OperationFailedReplication` en las Notificaciones de eventos de Amazon S3. Para obtener más información, consulte Recepción de eventos de error de replicación con notificaciones de eventos de Amazon S3.

Si un trabajo no se ejecuta en absoluto, las métricas no se envían a Amazon CloudWatch. Por ejemplo, su trabajo no se ejecutará si no tiene los permisos necesarios para ejecutar un trabajo de replicación por lotes de S3 o si las etiquetas o el prefijo de la configuración de replicación no coinciden.

Temas

- Habilitación de métricas de replicación de S3
- Recepción de eventos de error de replicación con notificaciones de eventos de Amazon S3

- Visualización de métricas de replicación con Almacenamiento de lente de S3
- Visualización de métricas de replicación mediante la consola de Amazon S3
- Motivos de errores de replicación de Amazon S3
- Obtención de información del estado de replicación

Habilitación de métricas de replicación de S3

Puede comenzar a utilizar métricas de replicación de S3 con una regla de replicación nueva o existente. Puede optar por aplicar la regla de replicación a un bucket de S3 completo o a objetos de Amazon S3 con un prefijo o etiqueta específicos.

Este tema proporciona instrucciones para habilitar las métricas de la replicación de S3 en la configuración de replicación cuando los buckets de origen y destino pertenecen a la misma o a otras Cuentas de AWS.

Para habilitar las métricas de replicación mediante la AWS Command Line Interface (AWS CLI), debe agregar una configuración de replicación al bucket de origen con `Metrics` habilitado. En esta configuración de ejemplo, los objetos con el prefijo *`Tax`* se replican en el bucket de destino *`DOC-EXAMPLE-BUCKET`* y se generan métricas de esos objetos.

```
{
    "Rules": [
        {
            "Status": "Enabled",
            "Filter": {
                "Prefix": "Tax"
            },
            "Destination": {
                "Bucket": "arn:aws:s3:::DOC-EXAMPLE-BUCKET",
                "Metrics": {
                    "Status": "Enabled"
                }
            },
            "Priority": 1
        }
    ],
    "Role": "IAM-Role-ARN"
}
```

Para obtener instrucciones completas sobre cómo crear reglas de replicación, consulte Configuración de la replicación para buckets de origen y destino que son propiedad de la misma cuenta.

Para obtener más información acerca de cómo ver métricas de replicación en la consola de S3, consulte Visualización de métricas de replicación mediante la consola de Amazon S3.

> **Note**
>
> Las métricas de replicación de S3 se facturan al mismo precio que las métricas personalizadas de Amazon CloudWatch. Para más información, consulte Precios de Amazon CloudWatch.

Recepción de eventos de error de replicación con notificaciones de eventos de Amazon S3

Las notificaciones de eventos de S3 pueden enviarle notificaciones mediante instancias cuando los objetos no se repliquen en el destino Región de AWS. Los eventos de Amazon S3 están disponibles a través de Amazon Simple Queue Service (Amazon SQS), Amazon Simple Notification Service (Amazon SNS) o AWS Lambda. Para obtener más información, consulte the section called "Notificaciones de eventos de Amazon S3".

Para obtener una lista de los códigos de error capturados por las notificaciones de eventos de S3, consulte Motivos de errores de replicación de Amazon S3.

Visualización de métricas de replicación con Almacenamiento de lente de S3

Para obtener métricas detalladas de la Replicación de S3, incluidas las métricas del recuento de reglas de replicación, puede utilizar la Lente de almacenamiento de Amazon S3. Lente de almacenamiento de S3 es una función de análisis de almacenamiento en la nube que puede utilizar para obtener visibilidad en toda la organización sobre el uso y la actividad del almacenamiento de objetos. Para obtener más información, consulte Uso de Lente de almacenamiento de S3 para proteger los datos. Para obtener una lista completa de las métricas, consulte el Glosario de métricas de Almacenamiento de lente de S3.

Visualización de métricas de replicación mediante la consola de Amazon S3

Existen tres tipos de métricas de Amazon CloudWatch para Amazon S3: métricas de almacenamiento, métricas de solicitud y métricas de replicación. Las métricas de replicación de S3

se activan automáticamente cuando se habilita la replicación con control del tiempo de replicación de S3 (S3 RTC) a través de la AWS Management Console o la API de Amazon S3. También puede habilitar las métricas de replicación de S3 independientemente del RTC de S3 al crear o editar una regla.

Las métricas de replicación realizan un seguimiento de los ID de regla de la configuración de replicación. Un ID de regla de replicación puede ser específico de un prefijo, de una etiqueta o de una combinación de ambos.

Para obtener más información acerca de las métricas de CloudWatch para Amazon S3, consulte Monitorización de métricas con Amazon CloudWatch.

Requisitos previos

Habilitar una regla de replicación con métricas de replicación de S3.

Para ver las métricas de replicación

1. Inicie sesión AWS Management Console Management Console y abra la consola de Amazon S3 en https://console.aws.amazon.com/s3/.
2. En el panel de navegación izquierdo, elija Instancias. En la lista Buckets, elija el nombre del bucket que contiene los objetos cuyas métricas de replicación desea obtener.
3. Elija la pestaña Metrics (Métricas).
4. En Replication metrics (Métricas de replicación), seleccione Replication rules (Reglas de replicación).
5. Elija Display charts (Mostrar gráficos).

   Amazon S3 muestra los gráficos Latencia de replicación (en segundos), Bytes pendientes de replicación, Operaciones pendientes de replicación y Operaciones en las que se produjo un error en la replicación.

A continuación podrá ver las métricas de replicación Latencia de replicación (en segundos), Operaciones pendientes de replicación, Bytes pendientes de replicación y Operaciones en las que se produjo un error en la replicación para las reglas que seleccione. Si utiliza Control de tiempo de replicación de S3, Amazon CloudWatch comienza a informar sobre las métricas de replicación 15 minutos después de activar S3 RTC en la regla de replicación respectiva. Puede ver métricas de replicación en la consola de Amazon S3 o de CloudWatch. Para obtener más información, consulte Métricas de replicación con S3 RTC.

> **Note**
>
> También puede ver métricas detalladas de la replicación de S3 en la consola de Amazon S3 con Lente de almacenamiento de Amazon S3. Lente de almacenamiento de S3 es una función de análisis de almacenamiento en la nube que puede utilizar para obtener visibilidad en toda la organización sobre el uso y la actividad del almacenamiento de objetos. Para obtener más información, consulte Uso de Lente de almacenamiento de S3 para proteger los datos. Para obtener una lista completa de las métricas, consulte el Glosario de métricas de Almacenamiento de lente de S3.

Motivos de errores de replicación de Amazon S3

En el siguiente gráfico se muestran los motivos de los errores de replicación de Amazon. Puede ver estos motivos al recibir el evento failureReason con las notificaciones de eventos de Amazon S3. Puede recibir notificaciones de eventos de S3 a través de Amazon Simple Queue Service (Amazon SQS), Amazon Simple Notification Service (Amazon SNS) o AWS Lambda. Para obtener más información, consulte Notificaciones de eventos de Amazon S3.

También puede ver estos motivos de error en un informe de finalización de replicación por lotes de S3. Para obtener más información, consulte Informe de finalización de replicación por lotes.

Motivo de error de replicación	Descripción
`AssumeRoleNotPermitted`	Amazon S3 no puede asumir el rol de AWS Identity and Access Management (IAM) especificado en la configuración de la replicación o en el trabajo de operaciones por lotes.
`DstBucketInvalidRegion`	El bucket de destino no está en la misma Región de AWS que la especificada en el trabajo de Operaciones por lotes. Este error es específico de la replicación por lotes.
`DstBucketNotFound`	Amazon S3 no encuentra el bucket de destino especificado en la configuración de replicación.

Motivo de error de replicación	Descripción
`DstBucketObjectLockConfigMissing`	Para replicar objetos de un bucket de origen con el bloqueo de objetos activado, el bucket de destino también debe tener habilitado el bloqueo de objetos. El error indica que es posible que el bloqueo de objetos no esté habilitado en el bucket de destino. Para obtener más información, consulte Consideraciones sobre el bloqueo de objetos.
`DstBucketUnversioned`	El control de versiones no está habilitado en el bucket de destino de S3. Habilite el control de versiones en el bucket de destino para replicar objetos con Replicación de S3.
`DstDelObjNotPermitted`	Amazon S3 no puede replicar los marcadores de eliminación en el bucket de destino. Es posible que falte el permiso `s3:ReplicateDelete` para el bucket de destino.
`DstKmsKeyInvalidState`	La clave de AWS Key Management Service (AWS KMS) del bucket de destino se encuentra en un estado no válido. Revise y habilite la clave AWS KMS requerida. Para obtener más información sobre la administración de claves de AWS KMS, consulte Estados de las claves de AWS KMS en la Guía para desarrolladores de AWS Key Management Service.
`DstKmsKeyNotFound`	La clave de AWS KMS clave configurada para el bucket de destino en la configuración de replicación no existe.

Motivo de error de replicación	Descripción
`DstMultipartCompleteNotPermitted`	Amazon S3 no puede completar la carga multiparte de objetos en el bucket de destino. Es posible que falte el permiso `s3:ReplicateObject` para el bucket de destino.
`DstMultipartInitNotPermitted`	Amazon S3 no puede iniciar la carga multiparte de objetos en el bucket de destino. Es posible que falte el permiso `s3:ReplicateObject` para el bucket de destino.
`DstMultipartPartUploadNotPermitted`	Amazon S3 no puede cargar objetos multiparte en el bucket de destino. Es posible que falte el permiso `s3:ReplicateObject` para el bucket de destino.
`DstObjectHardDeleted`	La replicación por lotes de S3 no admite volver a replicar objetos eliminados con el ID de versión del objeto del bucket de destino. Este error es específico de la replicación por lotes.
`DstPutAclNotPermitted`	Amazon S3 no puede replicar las listas de control de acceso (ACL) del objeto en el bucket de destino. Es posible que falte el permiso `s3:ReplicateObject` para el bucket de destino.
`DstPutLegalHoldNotPermitted`	Amazon S3 no puede aplicar una retención legal de bloqueo de objetos en los objetos de destino durante la replicación de objetos inmutables. Es posible que falte el permiso `s3:PutObjectLegalHold` para el bucket de destino. Para obtener más información, consulte Retenciones legales.

Motivo de error de replicación	Descripción
`DstPutObjectNotPermitted`	Amazon S3 no puede replicar objetos en el bucket de destino. Es posible que falten los permiso `s3:ReplicateObject` o `s3:ObjectOwnerOverrideToBucketOwner` para el bucket de destino.
`DstPutTaggingNotPermitted`	Amazon S3 no puede replicar etiquetas de objetos en el bucket de destino. Es posible que falte el permiso `s3:ReplicateObject` para el bucket de destino.
`DstVersionNotFound`	Amazon S3 no puede encontrar la versión del objeto requerida en el bucket de destino para la que se deben replicar los metadatos.
`InitiateReplicationNotPermitted`	Amazon S3 no puede iniciar la replicación en objetos. Puede que falte el permiso `s3:InitiateReplication` para el trabajo de operaciones por lotes. Este error es específico de la replicación por lotes.
`SrcBucketInvalidRegion`	El bucket de origen no está en la mismo Región de AWS que la especificada en el trabajo de operaciones por lotes. Este error es específico de la replicación por lotes.
`SrcBucketNotFound`	Amazon S3 no puede encontrar el bucket de origen.
`SrcBucketReplicationConfigMissing`	Amazon S3 no encuentra una configuración de replicación para el bucket de origen.

Motivo de error de replicación	Descripción
`SrcGetAclNotPermitted`	Amazon S3 no puede acceder al objeto del bucket de origen para la replicación. Es posible que falte el permiso `s3:GetObjectVersionAcl` para el objeto del bucket de origen. Los objetos del bucket de origen deben pertenecer al propietario del bucket. Si las ACL están habilitadas, compruebe si la propiedad del objeto está establecida en Propietario del bucket preferido o Escritor de objetos. Si la propiedad del objeto está establecida en Propietario del bucket preferido, los objetos del bucket de origen deben tener la ACL `bucket-owner-full-control` para que el propietario del bucket se convierta en propietario del objeto. La cuenta de origen puede asumir la propiedad de todos los objetos de su bucket configurando la propiedad de los objetos en Propietario del bucket obligatorio y desactivando las ACL.
`SrcGetLegalHoldNotPermitted`	Amazon S3 no puede acceder a la información de retención legal de Bloqueo de objetos de S3.
`SrcGetObjectNotPermitted`	Amazon S3 no puede acceder al objeto del bucket de origen para la replicación. Es posible que falte el permiso `s3:GetObjectVersionForReplication` para el bucket de origen.

Motivo de error de replicación	Descripción
`SrcGetRetentionNotPermitted`	Amazon S3 no puede acceder a la información del periodo de retención de Bloqueo de objetos de S3.
`SrcGetTaggingNotPermitted`	Amazon S3 no puede acceder a la información de las etiquetas de objetos desde el bucket de origen. Es posible que falte el permiso `s3:GetObjectVersionTagging` para el bucket de origen.
`SrcHeadObjectNotPermitted`	Amazon S3 no puede recuperar los metadatos del objeto del bucket de origen. Es posible que falte el permiso `s3:GetObjectVersionForReplication` para el bucket de origen.
`SrcKeyNotFound`	Amazon S3 no puede encontrar la clave del objeto de origen para replicarla. Es posible que el objeto de origen se haya eliminado antes de que se completara la replicación.
`SrcKmsKeyInvalidState`	La clave de AWS KMS del bucket de origen no tiene un estado válido. Revise y habilite la clave AWS KMS requerida. Para obtener más información sobre la administración de claves de AWS KMS, consulte Estados de las claves de AWS KMS en la Guía para desarrolladores de AWS Key Management Service.
`SrcObjectNotEligible`	Algunos objetos no son aptos para la replicación. Esto puede deberse a la clase de almacenamiento del objeto o a que las etiquetas del objeto no coinciden con la configuración de replicación.

Motivo de error de replicación	Descripción
`SrcObjectNotFound`	El objeto de origen no existe.
`SrcReplicationNotPending`	Amazon S3 ya ha replicado este objeto. Este objeto ya no está pendiente de replicación.
`SrcVersionNotFound`	Amazon S3 no puede encontrar la versión del objeto de origen para replicarla. Es posible que la versión del objeto de origen se haya eliminado antes de que se completara la replicación.

Temas relacionados de

Configuración de permisos para la replicación en directo

Solución de problemas de replicación

Obtención de información del estado de replicación

El estado de replicación puede ayudarle a determinar el estado actual de un objeto que se replica. El estado de replicación de un objeto de origen devolverá `PENDING`, `COMPLETED`, o `FAILED`. Se devolverá el estado de replicación de una réplica `REPLICA`.

Temas

- Información general sobre el estado de replicación
- Estado de replicación si se replica en varios buckets de destino
- Estado de replicación si la sincronización de modificación de réplica de Amazon S3 está habilitada
- Hallazgo del estado de replicación

Información general sobre el estado de replicación

En la replicación, tiene un bucket de origen en el que configurar la replicación y el destino donde Amazon S3 replica objetos. Cuando solicita un objeto (utilizando el objeto `GET`) o los metadatos de un objeto (utilizando el objeto `HEAD`) de estos buckets, Amazon S3 devuelve el encabezado `x-amz-replication-status` en la respuesta, como se indica a continuación:

- Al solicitar un objeto del bucket de origen, Amazon S3 devuelve el encabezado `x-amz-replication-status` si el objeto de su solicitud cumple los requisitos para la replicación.

  Por ejemplo, supongamos que en la configuración de replicación, usted especifica el prefijo del objeto `TaxDocs` en la configuración de replicación para indicar a Amazon S3 que replique objetos con el prefijo de nombre de clave `TaxDocs`. Cualquier objeto que cargue que tenga este prefijo de nombre de clave, por ejemplo, `TaxDocs/document1.pdf` se replicará. Para solicitudes de objetos con este prefijo de nombre de clave, Amazon S3 devuelve el encabezado `x-amz-replication-status` con uno de los siguientes valores para el estado de replicación del objeto: `PENDING`, `COMPLETED` o `FAILED`.

  > **Note**
  >
  > Si la replicación de objetos genera un error después de cargar un objeto, no puede volver a intentar la replicación. Deberá cargar de nuevo el objeto. Los objetos pasan a un estado `FAILED` para problemas como la falta de permisos de rol de replicación, permisos de AWS KMS o permisos de bucket. En el caso de los errores temporales, como si un bucket o región no están disponibles, el estado de replicación no pasará a `FAILED`, sino que permanecerá `PENDING`. Después de que el recurso vuelva a estar en línea, S3 reanudará la replicación de esos objetos.

- Cuando solicita un objeto desde un bucket de destino, si el objeto de la solicitud es una réplica creada por Amazon S3, este devuelve el `x-amz-replication-status` encabezado con el valor `REPLICA`.

> **Note**
>
> Antes de eliminar un objeto del bucket de origen que tiene activada la replicación, revise el estado de replicación del objeto para asegurarse de que el objeto haya sido replicado. Si la configuración del ciclo de vida está habilitada en el bucket de origen, Amazon S3 suspende las acciones del ciclo de vida hasta que marque el estado de los objetos como `COMPLETED` o bien `FAILED`.

Estado de replicación si se replica en varios buckets de destino

Cuando se replican objetos en varios buckets de destino, el `x-amz-replication-status` encabezado actúa de manera diferente. El encabezado del objeto de origen solo devuelve un valor de `COMPLETED` cuando la replicación se realiza correctamente en todos los destinos. El encabezado permanece en el `PENDING` valor hasta que se complete la replicación para todos los destinos. Si uno o más destinos fallan la replicación, el encabezado devuelve `FAILED`.

Estado de replicación si la sincronización de modificación de réplica de Amazon S3 está habilitada

Cuando las reglas de replicación habilitan la sincronización de las réplicas de modificación de Amazon S3, las réplicas pueden informar estados distintos de `REPLICA`. Si los cambios de metadatos están en proceso de replicación, el encabezado de `x-amz-replication-status` devuelve `PENDING`. Si la sincronización de modificaciones de réplica no permite replicar metadatos, el encabezado devuelve `FAILED`. Si los metadatos se replican correctamente, las réplicas devuelven el encabezado `REPLICA`.

Hallazgo del estado de replicación

Para obtener el estado de replicación de los objetos de un bucket, puede utilizar la herramienta de Amazon S3 Inventory. Amazon S3 envía un archivo CSV al bucket de destino que especifique en la configuración del inventario. También puede usar Amazon Athena para consultar el estado de replicación en el informe de inventario. Para obtener más información acerca del Amazon S3 Inventory, consulte Inventario de Amazon S3.

Puede buscar el estado de replicación de un objeto utilizando la consola, la AWS Command Line Interface (AWS CLI) o el SDK de AWS.

Uso de la consola de S3

En la consola de S3, puede ver el estado de replicación de un objeto en la página Detalles del objeto en Información general sobre la administración de objetos.

1. Inicie sesión AWS Management Console Management Console y abra la consola de Amazon S3 en https://console.aws.amazon.com/s3/.
2. En la lista de Buckets, seleccione el nombre del bucket.
3. En la lista Objects (Objetos), elija el nombre del objeto.
4. En la pestaña Properties (Propiedades), busque Object management overview (Descripción de administración de objetos), aquí podrá ver el estado de la replicación.

Uso de la AWS CLI

Utilice el `head-object` comando para recuperar metadatos de objeto, como se indica a continuación.

```
aws s3api head-object --bucket source-bucket --key object-key --version-id object-version-id
```

El comando devuelve los metadatos del objeto, incluido el `ReplicationStatus` como se muestra en el siguiente ejemplo de respuesta.

```
{
   "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":{

   }
}
```

Uso de los AWS SDK

Los siguientes fragmentos de código obtienen el estado de replicación con AWS SDK for Java y AWS SDK for .NET, respectivamente.

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);
```

Replicación de objetos existentes con replicación por lotes de S3

Al utilizar la replicación por lotes de S3, puede replicar los siguientes tipos de objetos:

- Objetos que existían antes de que se estableciera una configuración de replicación
- Objetos que se han replicado anteriormente
- Objetos que no se han podido replicar

Puede replicar estos objetos a demanda mediante un trabajo de operaciones por lotes. La replicación por lotes de S3 difiere de la replicación en directo que replica objetos nuevos de forma continua y automática en buckets de Amazon S3.

Para comenzar a utilizar la replicación por lotes, puede hacer lo siguiente:

- Iniciar la replicación por lotes para una nueva regla de replicación o destino: puede crear un trabajo de replicación por lotes único cuando cree la primera regla de una nueva configuración de replicación o agregue un nuevo destino a una configuración existente a través de la consola de Amazon S3.
- Iniciar la replicación por lotes para una configuración de replicación existente: puede crear un nuevo trabajo de replicación por lotes mediante Operaciones por lotes de S3 a través de la consola de Amazon S3, la AWS Command Line Interface (AWS CLI), los AWS SDK o la API de REST de Amazon S3.

Cuando termina el trabajo de replicación por lotes, recibe un reporte de finalización. Para obtener más información sobre cómo utilizar el reporte para examinar el trabajo, consulte Seguimiento del estado del trabajo e informes de finalización.

Consideraciones sobre la replicación por lotes de S3

- Su bucket de origen debe tener una configuración de replicación existente. Para habilitar la replicación, consulte Configuración de la replicación en directo y Ejemplos para configurar la replicación en directo.
- Si tiene configurado el ciclo de vida de S3 para su bucket, le recomendamos desactivar las reglas del ciclo de vida mientras el trabajo de replicación por lotes está activo. Al hacerlo le ayuda a garantizar la paridad entre los buckets de origen y de destino. De lo contrario, estos buckets podrían diferir y el bucket de destino no será una réplica exacta del bucket de origen. Por ejemplo, fíjese en el siguiente escenario:
  - El bucket de origen tiene varias versiones de un objeto y un marcador de eliminación en ese objeto.
  - Los buckets de origen y destino tienen una configuración de ciclo de vida para eliminar los marcadores de eliminación vencidos.

  En este caso, la replicación por lotes puede replicar el marcador de eliminación en el bucket de destino antes de replicar las versiones del objeto. Este comportamiento podría provocar que la configuración de ciclo de vida marcara el marcador de eliminación como caducado y el marcador de eliminación se eliminara del bucket de destino antes de replicar las versiones del objeto.
- El rol de AWS Identity and Access Management (IAM) que especifique para ejecutar el trabajo de la herramienta de Operaciones por lotes debe tener los permisos necesarios para realizar la operación subyacente de replicación por lotes. Para obtener más información sobre cómo crear un rol de IAM, consulte Configuración de políticas de IAM para replicación por lotes.
- La replicación por lotes requiere un manifiesto que Amazon S3 puede generar. El manifiesto generado debe almacenarse en la misma Región de AWS que el bucket de origen. Si elige no generar el manifiesto, puede proporcionar un informe de inventario de Amazon S3 o un archivo CSV que contenga los objetos que desea replicar.
- La replicación por lotes no admite volver a replicar objetos que se eliminaron con el ID de versión del objeto del bucket de destino. Para volver a replicar estos objetos, puede copiar los objetos de origen en su lugar con un trabajo de copia por lotes. Al copiar esos objetos en su lugar, se crean nuevas versiones de los objetos en el bucket de origen e inicia la replicación automáticamente en el bucket de destino. Al eliminar y volver a crear el bucket de destino no se inicia la replicación.

  Para obtener más información acerca de la copia por lotes, consulte Ejemplos donde se utilizan las operaciones por lotes para copiar objetos.

- Si utiliza una regla de replicación en el bucket de S3, asegúrese de actualizar la configuración de replicación y conceder al rol de IAM asociado a la regla de replicación los permisos adecuados para replicar objetos. El rol de IAM debe tener permisos para realizar la replicación tanto en el bucket de origen como en el de destino.
- Si envía varios trabajos de replicación por lotes para el mismo bucket en un periodo breve, Amazon S3 ejecutará esos trabajos simultáneamente.
- Si envía varios trabajos de replicación por lotes para dos buckets diferentes, tenga en cuenta que es posible que Amazon S3 no ejecute todos los trabajos simultáneamente. Si supera el número de trabajos de replicación por lotes que se pueden ejecutar a la vez en su cuenta, Amazon S3 detendrá los trabajos de menor prioridad para trabajar en los de mayor prioridad. Cuando se hayan completado los elementos de mayor prioridad, todos los trabajos en pausa volverán a estar activos.
- La replicación por lotes no es compatible para los objetos almacenados en las clases de almacenamiento S3 Glacier Flexible Retrieval y S3 Glacier Deep Archive.
- Para replicar por lotes objetos S3 Intelligent-Tiering almacenados en el nivel de almacenamiento Acceso a archivos o Acceso a archivos profundo, primero debe iniciar una solicitud de restauración y esperar a que los objetos se muevan al nivel Acceso frecuente.

Especificación de un manifiesto para un trabajo de replicación por lotes

Un manifiesto es un objeto de Amazon S3 que contiene las claves de objeto sobre las que desea que actúe Amazon S3. Si desea crear un trabajo de replicación por lotes, debe proporcionar un manifiesto generado por el usuario o hacer que Amazon S3 genere un manifiesto en función de la configuración de replicación.

Si proporciona un manifiesto generado por el usuario, debe ser en forma de informe de inventario de Amazon S3 o un archivo CSV. Si los objetos del manifiesto están en un bucket con control de versiones, debe especificar los ID de versión de los objetos. Solo se replicará el objeto con el ID de versión especificado en el manifiesto. Para obtener más información sobre cómo especificar un manifiesto, consulte Especificar un manifiesto.

Si elige que Amazon S3 genere un archivo de manifiesto en su nombre, los objetos enumerados utilizarán el mismo bucket de origen, el mismo prefijo y las mismas etiquetas que todas las configuraciones de replicación del bucket de origen. Con un manifiesto generado, Amazon S3 replicará todas las versiones aptas de sus objetos.

> **Note**
>
> Si elige que Amazon S3 genere el manifiesto, este debe almacenarse en la misma Región de AWS que el bucket de origen.

Filtros para un trabajo de replicación por lotes

En el momento de crear el trabajo de replicación por lotes, tiene la opción de especificar filtros adicionales, como la fecha de creación de los objetos y el estado de replicación para reducir el alcance del trabajo.

Puede filtrar los objetos para replicar en función del valor `ObjectReplicationStatuses`, proporcionando uno o varios de los siguientes valores:

- `"NONE"`: indica que Amazon S3 nunca intentó replicar el objeto antes.
- `"FAILED"`: indica que Amazon S3 intentó replicar el objeto antes, pero no pudo.
- `"COMPLETED"`: indica que Amazon S3 replicó el objeto correctamente antes.
- `"REPLICA"`: indica que se trata de un objeto de réplica que Amazon S3 ha replicado desde otro origen.

Para obtener más información sobre los estados de replicación, consulte Obtención de información del estado de replicación.

Si no filtra su trabajo de replicación por lotes, Operaciones por lotes intentará replicar todos los objetos (independientemente de cuáles sean sus `ObjectReplicationStatus`) en el manifiesto que coincida con las reglas de la configuración de replicación, excepto algunos objetos que no se replican de forma predeterminada. Para obtener más información, consulte the section called “¿Qué elementos no se replican con las configuraciones de replicación?”

Dependiendo de su objetivo, puede establecer `ObjectReplicationStatuses` en uno o más de los siguientes valores:

- Para replicar solo los objetos existentes que nunca se han replicado, incluya únicamente `"NONE"`.
- Para volver a intentar replicar solo los objetos que no se habían podido replicar antes, incluya únicamente `"FAILED"`.
- Para replicar objetos existentes y volver a intentar replicar objetos que no se pudieron replicar anteriormente, incluya tanto `"NONE"` como `"FAILED"`.

- Para reponer un bucket de destino con objetos que se han replicado en otro destino, incluya `"COMPLETED"`.
- Para replicar objetos previamente replicados, incluya `"REPLICA"`.

Informe de finalización de replicación por lotes

Al crear un trabajo de replicación por lotes, puede solicitar un informe de finalización CSV. Este informe muestra los objetos, los códigos de éxito o error de la replicación, las salidas y las descripciones. Para obtener más información sobre el seguimiento del trabajo y los informes de finalización, consulte Informes de finalización.

Para obtener una lista de códigos y descripciones de errores de replicación, consulte Motivos de errores de replicación de Amazon S3.

Para obtener más información sobre la resolución de problemas de la replicación por lotes, consulte Errores de replicación por lotes.

Introducción a la replicación por lotes

Para obtener más información sobre cómo utilizar la replicación por lotes, consulte el Tutorial: Replicación de objetos existentes en los buckets de Amazon S3 con la replicación por lotes de S3.

Configuración de políticas de IAM para replicación por lotes

Dado que la replicación por lotes de S3 es un tipo de trabajo de operaciones por lote, debe crear un rol de AWS Identity and Access Management (IAM) de operaciones por lote a fin de conceder permisos de Amazon S3 para realizar acciones en su nombre. También debe adjuntar una política de IAM de replicación por lotes al rol de IAM de operaciones por lotes. En el ejemplo siguiente, se crea un rol de IAM que otorga permiso a las operaciones por lotes para iniciar un trabajo de replicación por lotes.

Creación de una política y un rol de IAM

1. Inicie sesión en AWS Management Console Management Console y abra la consola IAM en https://console.aws.amazon.com/iam/.
2. En Access management (Administración de acceso), elija Roles (Roles).
3. Elija Create Role (Crear rol).

4. Elija Servicio de AWS como tipo de entidad de confianza, Amazon S3 como servicio y S3 Batch Operations (Operaciones por lotes de S3) como caso de uso.
5. Elija Next: Permissions (Siguiente: permisos).
6. Elija Create Policy (Crear política).
7. Elija JSON e inserte una de las siguientes políticas en función del manifiesto.

> **Note**
>
> Se necesita un permiso diferente si está generando un manifiesto o suministrando uno. Para obtener más información, consulte, Especificación de un manifiesto para un trabajo de replicación por lotes.

Política en caso de que se utilice y almacene un manifiesto generado por 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 ****/*"
         ]
      }
   ]
}
```

Política en caso de que se utilice un manifiesto proporcionado por el usuario

```
{
   "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. Elija Next: Tags (Siguiente: etiquetas).
9. Elija Next: Review (Siguiente: revisar).
10. Elija un nombre para la política y elija Create policy (Crear política).
11. Adjunte esta política a su rol y elija Next: Tags (Siguiente: Etiquetas).
12. Elija Next: Review (Siguiente: revisar).
13. Elija un nombre para el rol y, a continuación, elija Create role (Crear rol).

Verificación de la política de confianza

1. Inicie sesión en AWS Management Console Management Console y abra la consola IAM en https://console.aws.amazon.com/iam/.
2. En Access management (Administración de acceso), elija Roles y seleccione el rol recién creado.
3. En la pestaña Trust relationships (Relaciones de confianza), elija Edit trust relationship (Editar relación de confianza).
4. Verifique que este rol utilice la siguiente política de confianza:

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

```
        }
    ]
}
```

Creación de un trabajo de replicación por lotes para una primera regla de replicación o un nuevo destino

Cuando se crea la primera regla en una nueva configuración de replicación o cuando se agrega un nuevo destino a una configuración existente mediante la AWS Management Console, puede crear opcionalmente un trabajo de replicación por lotes.

Para utilizar la replicación por lotes para una configuración existente sin agregar un nuevo destino, consulte Creación de un trabajo de replicación por lotes para las reglas de replicación existentes.

Uso de la replicación por lotes para una nueva regla de replicación o destino a través de la AWS Management Console

1. Inicie sesión AWS Management Console Management Console y abra la consola de Amazon S3 en https://console.aws.amazon.com/s3/.
2. En la lista Buckets (Buckets), elija el nombre del bucket que contiene los objetos que desea replicar.
3. Para crear una nueva regla de replicación o editar una existente, elija Management (Administración) y desplácese hacia abajo hasta Replication rules (Reglas de replicación):

   - Para crear una nueva regla de replicación, elija Create replication rule (Crear regla de replicación).

     > **Note**
     >
     > Para obtener ejemplos sobre cómo configurar una regla de replicación básica, consulte Ejemplos para configurar la replicación en directo.

   - Para editar una regla de replicación existente, seleccione la regla y, a continuación, elija Edit rule (Editar regla).
4. Cree la nueva regla de replicación o edite el destino de la regla de replicación existente y elija Save (Guardar).

Después de crear la primera regla en una nueva configuración de replicación o de editar una configuración existente para agregar un nuevo destino, aparece un cuadro de diálogo Replicate existing objects? (¿Replicar objetos existentes?), que le da la opción de crear un trabajo de replicación por lotes.

5. Si desea ejecutar este trabajo ahora, elija Sí, replicar los objetos existentes.

   Alternativamente, si elige No, no replicar objetos existentes puede ejecutar este trabajo más adelante.

6. Crear el trabajo de replicación por lotes de S3. El trabajo de replicación por lotes de S3 tiene varias configuraciones:

   Opción de ejecución de trabajo

   Si desea que el trabajo de replicación por lotes de S3 se ejecute de inmediato, puede elegir Job runs automatically when ready (El trabajo se ejecuta automáticamente cuando está listo). Si desea ejecutar el trabajo más adelante, elija Job waits to be run when ready (El trabajo espera a ejecutarse cuando esté listo).

   Si elige Job waits to be run when ready (Trabajo en espera hasta que esté listo para ser ejecutado), no podrá crear ni guardar un manifiesto de operaciones por lotes. Para guardar el manifiesto de operaciones por lotes, elija Job waits to be run when ready (Trabajo en espera hasta que esté listo para ser ejecutado).

   Manifiesto de operaciones por lotes

   El manifiesto es una lista de todos los objetos en los que quiere que se ejecute la acción especificada. Puede elegir guardar el manifiesto de operaciones por lotes. Al igual que en los archivos de inventario de S3, el manifiesto se guardará como archivo CSV y se almacenará en un bucket. Para obtener más información sobre los manifiestos de operaciones por lotes, consulte Especificar un manifiesto.

   Reporte de finalización

   Las operaciones por lotes de S3 ejecutan una tarea para cada objeto especificado en el manifiesto. Los informes de finalización constituyen un mecanismo sencillo para ver los resultados de las tareas en un formato unificado sin necesidad de realizar ninguna configuración adicional. Puede solicitar un reporte de finalización para todas las tareas o solo

para las tareas que fallaron. Para obtener más información sobre los reportes de finalización, consulte Informes de finalización.

Permisos

Una de las causas más comunes de los errores de replicación son los permisos insuficientes en el rol de AWS Identity and Access Management (IAM). Para obtener información sobre la creación de este rol, consulte Configuración de políticas de IAM para replicación por lotes.

7. Elija Create Batch Operations job (Crear trabajo de operaciones por lotes).

Creación de un trabajo de replicación por lotes para las reglas de replicación existentes

Puede configurar la replicación por lotes de S3 para una configuración de replicación existente mediante los AWS SDK, la AWS Command Line Interface (AWS CLI) o la consola de Amazon S3. Para obtener información general sobre la replicación por lotes, consulte Replicación de objetos existentes con replicación por lotes de S3.

Como requisito previo, debe crear un rol de AWS Identity and Access Management (IAM) para operaciones por lotes a fin de otorgar permisos de Amazon S3 para realizar acciones en su nombre. Consulte Configuración de políticas de IAM para replicación por lotes.

Cuando termina el trabajo de replicación por lotes, recibe un reporte de finalización. Para obtener más información sobre cómo utilizar el reporte para examinar el trabajo, consulte Seguimiento del estado del trabajo e informes de finalización.

Uso de la consola de S3

1. Inicie sesión en la AWS Management Console y abra la consola de Amazon S3 en https://console.aws.amazon.com/s3/.
2. Seleccione Batch Operations (Operaciones por lotes) en el panel de navegación de la consola de Amazon S3.
3. Seleccione Create job (Crear trabajo).
4. Elija en Region (Región) la región en la que desea crear el trabajo.
5. Seleccione el Manifest format (Formato del manifiesto). En este ejemplo, se mostrará cómo crear un manifiesto basado en una configuración de replicación de S3 existente.

> **Note**
>
> El manifiesto es una lista de todos los objetos en los que quiere que se ejecute la acción especificada. Para obtener más información sobre los manifiestos de operaciones por lotes, consulte Especificar un manifiesto. Si tiene preparado un manifiesto, elija S3 inventory report (manifest.json) (Reporte de inventario de S3 [manifest.json]) o CSV. Si los objetos del manifiesto están en un bucket con control de versiones, debe especificar los ID de versión de los objetos. Para obtener más información sobre la creación de un manifiesto, consulte Especificar un manifiesto.

6. Para crear un manifiesto basado en la configuración de replicación, elija Create manifest using S3 Replication configuration (Crear manifiesto mediante la configuración de replicación de S3). A continuación, elija el bucket de origen de la configuración de replicación.
7. (Opcional) Puede incluir filtros adicionales, como la fecha de creación de objetos y el estado de replicación. Para obtener ejemplos sobre cómo filtrar por estado de replicación, consulte Especificación de un manifiesto para un trabajo de replicación por lotes.
8. Para guardar un manifiesto, seleccione Save Batch Operations manifest (Guardar el manifiesto de operaciones por lote).

   a. Si decide generar y guardar un manifiesto, debe elegir Bucket in this account (Bucket en esta cuenta) o Bucket in another Cuenta de AWS (Bucket en otra Cuenta de AWS). Especifique el nombre del bucket en el cuadro de texto.

   > **Note**
   >
   > El manifiesto generado debe almacenarse en la misma Región de AWS que el bucket de origen.

   b. Elija el Tipo de cifrado.
9. (Opcional) Proporcione una Description (Descripción).
10. Ajuste la Priority (Prioridad) del trabajo si es necesario. Los números más elevados indican mayor prioridad. Amazon S3 intenta ejecutar los trabajos de mayor prioridad antes que los trabajos de menor prioridad. Para obtener más información acerca de la prioridad de un trabajo, consulte Asignar prioridad a los trabajos.
11. (Opcional) Genere un reporte de finalización. Para generar el reporte, seleccione Generate completion report (Generar reporte de finalización).

Si elige generar un reporte de finalización, debe elegir informar Failed tasks only (Solo tareas con errores) o All tasks (Todas las tareas) y proporcionar un bucket de destino para el reporte.

12. Seleccione un rol de IAM válido.

> **Note**
>
> Para obtener más información sobre cómo crear un rol de IAM, consulte Configuración de políticas de IAM para replicación por lotes.

13. (Opcional) Agregue etiquetas de trabajo al trabajo de replicación por lotes.
14. Elija Next (Siguiente).
15. Revise la configuración del trabajo y seleccione Create job (Crear trabajo).

Uso de la AWS CLI con un manifiesto de S3

En el siguiente ejemplo, se crea un trabajo de replicación por lotes de S3 mediante un manifiesto generado por S3 para la Cuenta de AWS *111122223333*. En este ejemplo, se intentará replicar objetos existentes y objetos que no se habían podido replicar anteriormente. Para obtener información sobre el filtrado por estado de replicación, consulte Especificación de un manifiesto para un trabajo de replicación por lotes.

```
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**
>
> El trabajo debe iniciarse desde la misma Región de AWS del bucket de replicación de origen. El rol de IAM `role/batch-Replication-IAM-policy` se creó anteriormente. Consulte Configuración de políticas de IAM para replicación por lotes.

Después de iniciar correctamente un trabajo de replicación por lotes, recibirá el ID del trabajo como respuesta. Puede supervisar este trabajo con el siguiente comando.

```
aws s3control describe-job --account-id 111122223333 --job-id job-id --region source-
bucket-region
```

Uso de la AWS CLI con un manifiesto proporcionado por el usuario

En el siguiente ejemplo, se crea un trabajo de replicación por lotes de S3 mediante un manifiesto definido por el usuario para la Cuenta de AWS *111122223333*. Si los objetos del manifiesto están en un bucket con control de versiones, debe especificar los ID de versión de los objetos. Solo se replicará el objeto con el ID de versión especificado en el manifiesto. Para obtener más información sobre la creación de un manifiesto, consulte Especificar un manifiesto.

```
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**
>
> El trabajo debe iniciarse desde la misma Región de AWS del bucket de replicación de origen. El rol de IAM `role/batch-Replication-IAM-policy` se creó anteriormente. Consulte Configuración de políticas de IAM para replicación por lotes.

Después de iniciar correctamente un trabajo de replicación por lotes, recibirá el ID del trabajo como respuesta. Puede supervisar este trabajo con el siguiente comando.

```
aws s3control describe-job --account-id 111122223333 --job-id job-id --region source-
bucket-region
```

Categorización del almacenamiento mediante etiquetas

Utilice el etiquetado de objetos para categorizar el almacenamiento. Cada etiqueta es un par clave-valor.

Puede agregar etiquetas a nuevos objetos al cargarlos o agregarlas a objetos existentes.

- Puede asociar hasta 10 etiquetas a un objeto. Las etiquetas que están asociadas con un objeto deben tener claves de etiquetas exclusivas.
- Una clave de etiqueta puede tener una longitud de hasta 128 caracteres Unicode y los valores de etiqueta pueden tener una longitud de hasta 256 caracteres Unicode. Las etiquetas de objetos de Amazon S3 se representan internamente en UTF-16. Tenga en cuenta que en UTF-16, los caracteres ocupan 1 o 2 posiciones de caracteres.
- La clave y los valores distinguen entre mayúsculas y minúsculas.
- Para obtener más información sobre las restricciones en las etiquetas, consulte Restricciones de las etiquetas definidas por el usuario en la Guía del usuario de Administración de facturación y costos de AWS. Para obtener información sobre las restricciones básicas de las etiquetas, consulte Restricciones de etiquetas en la Guía del usuario de Amazon EC2.

Ejemplos

Considere los siguientes ejemplos de etiquetado:

Example Información sanitaria protegida (PHI)

Suponga que un objeto contiene datos de información sanitaria protegida (PHI). Podría etiquetar el objeto con el siguiente par clave-valor.

```
PHI=True
```

o

```
Classification=PHI
```

Example Archivos de proyecto

Supongamos que almacena archivos de proyecto en su bucket de S3. Podría etiquetar estos objetos con una clave llamada `Project` y un valor, como se muestra a continuación:

```
Project=Blue
```

Example Múltiples etiquetas

Puede agregar varias etiquetas a un objeto, como se muestra a continuación.

```
Project=x
Classification=confidential
```

Prefijos y etiquetas de nombre de clave

Los prefijos de nombre de clave de objeto también le permiten clasificar el almacenamiento. Sin embargo, la categorización basada en prefijos es unidimensional. Considere los siguientes nombres de claves de objeto:

```
photos/photo1.jpg
project/projectx/document.pdf
project/projecty/document2.pdf
```

Estos nombres de claves tienen los prefijos `photos/`, `project/projectx/` y `project/projecty/`. Estos prefijos habilitan la categorización monodimensional. Es decir, todo lo que tenga un mismo prefijo es una categoría. Por ejemplo, el prefijo `project/projectx` identifica a todos los documentos relacionados con project x.

Con el etiquetado, ahora tendrá otra dimensión. Si quiere que la foto 1 esté en la categoría project x, puede etiquetar el objeto correspondientemente.

Beneficios adicionales

Además de la clasificación de datos, el etiquetado ofrece los siguientes beneficios:

- Las etiquetas de objetos permiten el control de acceso pormenorizado para otorgar permisos. Por ejemplo, podría conceder a un usuario permisos para leer solo objetos con determinadas etiquetas.
- Las etiquetas de objetos permiten una administración precisa del ciclo de vida de un objeto, en la que podrá especificar filtros basados en etiquetas, además de prefijos de nombres de clave, en una regla de ciclo de vida.
- Cuando utilice los análisis de Amazon S3 puede configurar filtros para agrupar los objetos de modo que se analicen por etiquetas de objetos, por prefijos de nombre de clave o por prefijos y etiquetas.

- También puede personalizar métricas de Amazon CloudWatch para mostrar información especificando filtros de etiquetas. En las siguientes secciones presentamos más detalles.

> **Important**
>
> Es aceptable usar etiquetas para etiquetar objetos que contengan información confidencial (como información personalmente identificable o información sanitaria protegida). No obstante, las etiquetas en sí no deberían contener información confidencial.

Agregar conjuntos de etiquetas de objetos a varios objetos de Amazon S3 con una sola solicitud.

Para añadir conjuntos de etiquetas de objetos á más de un objeto de Amazon S3 con una sola solicitud, puede utilizar Operaciones por lotes de S3. Proporcione a Operaciones por lotes de S3 una lista de objetos en los que operar. Operaciones por lotes de S3 llama a la operación de la API respectiva para realizar la operación especificada. Un solo trabajo de operaciones por lotes puede realizar la operación especificada en miles de millones de objetos con exabytes de datos.

Operaciones por lotes de S3 realiza un seguimiento del avance, envía notificaciones y guarda un informe de finalización de todas las acciones, por lo que proporciona una experiencia sin servidor, auditable, completamente administrada. Puede emplear Operaciones por lotes de S3 a través de la consola de Amazon S3, AWS CLI, los SDK de AWS o la API de REST. Para obtener más información, consulte the section called “Conceptos básicos de las operaciones por lotes”.

Para obtener más información acerca de las etiquetas de objeto, consulte Administrar etiquetas de objetos.

Operaciones de la API relacionadas con el etiquetado de objetos

Amazon S3 admite las siguientes operaciones de la API específicas al etiquetado de objetos:

Operaciones de la API para objetos

- PUT Object tagging: sustituye las etiquetas en un objeto. Las etiquetas se especifican en el cuerpo de la solicitud. Hay dos escenarios diferentes de administración de etiquetas de objetos con esta API.
  - Si el objeto no tiene etiquetas: con esta API puede agregar un conjunto de etiquetas a un objeto (el objeto no tiene etiquetas anteriores).

- Si el objeto tiene un conjunto de etiquetas existente: para modificar el conjunto de etiquetas existente, en primer lugar debe recuperar el conjunto de etiquetas existente, modificarlo en el cliente y usar esta API para sustituir el conjunto de etiquetas.

  > **Note**
  >
  > Si envía esta solicitud con un conjunto de etiquetas vacío, Amazon S3 elimina el conjunto de etiquetas existente en el objeto. Si utiliza este método, se le cobrará por una solicitud de nivel 1 (PUT). Para obtener más información, consulte Precios de Amazon S3.
  >
  > La solicitud DELETE Object tagging es la preferida porque consigue el mismo resultado sin incurrir en cargos.

- GET Object tagging: devuelve el conjunto de etiquetas asociado con un objeto. Amazon S3 devuelve las etiquetas de objeto en el cuerpo de la respuesta.
- DELETE Object tagging: elimina el conjunto de etiquetas asociadas con un objeto.

Otras operaciones de la API que admiten etiquetado

- PUT Object e Initiate Multipart Upload: puede especificar etiquetas al crear los objetos. Las etiquetas se especifican con el encabezado de solicitud `x-amz-tagging`.
- GET Object: en lugar de devolver el conjunto de etiquetas, Amazon S3 devuelve el recuento de etiquetas de objeto en el encabezado `x-amz-tag-count` (solo si el solicitante tiene permiso para leer las etiquetas) dado que el tamaño del encabezado de respuesta está limitado a 8 Kb. Si quiere ver las etiquetas, realice otra solicitud para la operación de la API GET Object tagging.
- POST Object: puede especificar las etiquetas en su solicitud POST.

  Siempre que las etiquetas de su solicitud no excedan el límite de tamaño de 8 Kb para los encabezados de solicitud, puede usar la API `PUT Object` para crear objetos con etiquetas. Si las etiquetas que especifique superan el límite de tamaño del encabezado, puede usar este método POST, en el que incluiría las etiquetas en el cuerpo.

  PUT Object - Copy: puede especificar el `x-amz-tagging-directive` en su solicitud para dar la instrucción a Amazon S3 de que copie (comportamiento predeterminado) las etiquetas o sustituya por un nuevo conjunto de etiquetas facilitadas en la solicitud.

Tenga en cuenta lo siguiente:

- El etiquetado de objetos S3 es muy consistente. Para obtener más información, consulte Modelo de consistencia de datos de Amazon S3.

Configuraciones adicionales

En esta sección, se explica cómo el etiquetado de objetos se relaciona con otras configuraciones.

Etiquetar objetos y administrar el ciclo de vida

En la configuración del ciclo de vida del bucket, puede especificar un filtro para seleccionar un subconjunto de objetos al que se aplica la regla. Puede especificar un filtro en función de los prefijos de nombres de clave, etiquetas de objetos o ambos.

Supongamos que almacena fotos (en formato bruto y terminado) en su bucket de Amazon S3. Puede etiquetar estos objetos como se muestra a continuación.

```
phototype=raw
or
phototype=finished
```

Podría plantearse archivar las fotos brutas en S3 Glacier tiempo después de que se creen. Puede configurar una regla de ciclo de vida con un filtro que identifique el subconjunto de objetos con el prefijo de nombre de clave (`photos/`) que tiene una etiqueta específica (`phototype=raw`).

Para obtener más información, consulte Administración del ciclo de vida del almacenamiento.

Etiquetar y replicar objetos

Si configura la replicación en un bucket, Amazon S3 replica las etiquetas, siempre que conceda permisos a Amazon S3 para leer las etiquetas. Para obtener más información, consulte Configuración de la replicación en directo.

Notificaciones de eventos de etiquetado de objetos

Puede configurar una notificación de evento de Amazon S3 para recibir un aviso cuando se agrega o elimina una etiqueta de un objeto. El tipo de evento `s3:ObjectTagging:Put` le notifica cuando una etiqueta es PUT en un objeto o cuando se actualiza una etiqueta existente. El tipo de evento

`s3:ObjectTagging:Delete` le notifica cuando se quita una etiqueta de un objeto. Para obtener más información, consulte Habilitación de notificaciones de eventos.

Para obtener más información sobre el etiquetado de objetos, vea los siguientes temas:

Temas

- Etiquetado y políticas de control de acceso
- Administrar etiquetas de objetos

Etiquetado y políticas de control de acceso

También puede usar políticas de permisos (políticas de bucket y usuario) para administrar los permisos relacionados con el etiquetado de objetos. Para ver acciones de políticas, consulte los siguientes temas:

- Operaciones con objetos
- Operaciones con buckets

Las etiquetas de objetos permiten un control de acceso pormenorizado para administrar permisos. Puede otorgar permisos condicionales en función de las etiquetas de objetos. Amazon S3 admite las siguientes claves de condiciones que puede usar para conceder permisos condicionales basados en etiquetas de objetos.

- `s3:ExistingObjectTag/<tag-key>`: use esta clave condicional para verificar una etiqueta de objeto existente tiene una clave y un valor específicos para la etiqueta.

  > **Note**
  >
  > Al conceder permisos para las operaciones `PUT Object` y `DELETE Object`, esta clave condicional no se admite. Es decir, no puede crear una política para conceder o denegar permisos a un usuario para eliminar o sobrescribir un objeto en función de sus etiquetas existentes.

- `s3:RequestObjectTagKeys`: use esta clave condicional para restringir las claves de etiqueta que quiera permitir en objetos. Esto resulta útil al agregar etiquetas a objetos con PutObjectTagging y PutObject y con las solicitudes POST para objetos.

- `s3:RequestObjectTag/<tag-key>`: use esta clave condicional para restringir las claves y valores de etiqueta que quiera permitir en objetos. Esto resulta útil al agregar etiquetas a objetos con PutObjectTagging y PutObject y con las solicitudes POST para buckets.

Para obtener una lista completa de las claves condicionales específicas de servicio de Amazon S3, consulte Ejemplos de políticas de bucket que utilizan claves de condición. Las siguientes políticas de permisos ilustran cómo el etiquetado de objetos facilita una administración de permisos de acceso pormenorizada.

Example 1: Permitir a un usuario leer solo los objetos que tienen una clave y valor de etiqueta específicos

La siguiente política de permisos limita al usuario a leer solo los objetos que tengan la clave y el valor de la etiqueta `environment: production`. Esta política usa la clave de condición `s3:ExistingObjectTag` para especificar la clave y el valor de etiqueta.

```
{
  "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: Restringir las claves de etiqueta de objetos que los usuarios pueden agregar

La siguiente política de permisos concede permisos a un usuario para realizar la acción `s3:PutObjectTagging`, lo que permite al usuario agregar etiquetas a un objeto existente. La condición usa la clave de condición `s3:RequestObjectTagKeys` para especificar las claves

de etiqueta permitidas, como `Owner` o `CreationDate`. Para obtener más información, consulte Creación de una condición que pruebe valores de varias claves en la Guía para usuarios de IAM.

La política garantiza que cada clave de etiqueta especificada en la solicitud sea una clave de etiqueta autorizada. El calificador `ForAnyValue` de la condición garantiza que al menos una de las claves especificadas estará presente en la solicitud.

```
{
   "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: Requerir una clave y un valor de etiqueta específica al permitir a los usuarios agregar etiquetas de objetos

La política de ejemplo siguiente concede un permiso de usuario para realizar la acción `s3:PutObjectTagging`, lo que permite al usuario agregar etiquetas a un objeto existente. La condición requiere que el usuario incluya una clave de etiqueta específica (como *`Project`*) con el valor establecido en *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"
        }
      }
    }
  ]
}
```

Administrar etiquetas de objetos

En esta sección, se explica cómo puede administrar etiquetas de objetos con los SDK de AWS para Java y .NET, o la consola de Amazon S3.

El etiquetado de objetos le permite categorizar el almacenamiento. Cada etiqueta es un par clave-valor que se ajusta a las reglas siguientes:

- Puede asociar hasta 10 etiquetas a un objeto. Las etiquetas que están asociadas con un objeto deben tener claves de etiquetas exclusivas.
- Una clave de etiqueta puede tener una longitud de hasta 128 caracteres Unicode y los valores de etiqueta pueden tener una longitud de hasta 256 caracteres Unicode. Las etiquetas de objetos de Amazon S3 se representan internamente en UTF-16. Tenga en cuenta que en UTF-16, los caracteres ocupan 1 o 2 posiciones de caracteres.
- La clave y los valores distinguen entre mayúsculas y minúsculas.

Para obtener más información acerca de las etiquetas de objeto, consulte Categorización del almacenamiento mediante etiquetas. Para obtener más información sobre las restricciones de las etiquetas, consulte Restricciones de las etiquetas definidas por el usuario en la Guía del usuario de AWS Billing and Cost Management.

Uso de la consola de S3

Para añadir etiquetas a un objeto

1. Inicie sesión en la AWS Management Console y abra la consola de Amazon S3 en https://console.aws.amazon.com/s3/.
2. En la lista Buckets, seleccione el nombre del bucket que contiene los objetos a los que desea agregar etiquetas.

   Si lo desea, también puede ir una carpeta.
3. En la lista Objects (Objetos), seleccione la casilla de verificación situada junto a los nombres de los objetos a los que desea agregarles etiquetas.
4. En el menú Actions (Acciones), elija Edit (Editar).
5. Revise los objetos mencionados y elija Add tags (Agregar etiquetas).
6. Cada etiqueta de objeto es un par clave-valor. Introduzca la información pertinente en Key (Clave) y Value (Valor). Para agregar otra etiqueta, elija Add Tag (Añadir etiqueta).

   Puede introducir hasta 10 etiquetas para un objeto.
7. Elija Save changes.

   Amazon S3 agrega las etiquetas a los objetos especificados.

Para obtener más información, consulte Visualización de propiedades de objeto en la consola de Amazon S3 y Carga de objetos en esta guía.

Uso de la SDKs AWS

Java

En la siguiente muestra se indica cómo utilizar el AWS SDK for Java para establecer etiquetas para un nuevo objeto y recuperar o reemplazar etiquetas de un objeto ya existente. Para obtener más información acerca de cómo etiquetar objetos, consulte Categorización del almacenamiento mediante etiquetas. Para obtener instrucciones sobre cómo crear y probar una muestra funcional, consulte Introducción en la Guía del desarrollador de AWS SDK for Java.

```
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

En la siguiente muestra se indica cómo utilizar el AWS SDK for .NET para establecer las etiquetas para un nuevo objeto y recuperar o reemplazar las etiquetas de un objeto ya existente. Para obtener más información acerca de cómo etiquetar objetos, consulte Categorización del almacenamiento mediante etiquetas.

Para obtener información acerca de cómo configurar y ejecutar ejemplos de código, consulte Introducción al SDK de AWS para .NET en la Guía para desarrolladores del SDK de AWS para .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);
            }
        }
    }
}
```

Uso de etiquetas de buckets de S3 de asignación de costos

Para realizar un seguimiento del costo de almacenamiento u otros criterios para proyectos individuales o grupos de proyectos, etiquete los buckets de Amazon S3 con etiquetas de asignación de costos. Una etiqueta de asignación de costos es un par clave-valor que se asocia a un bucket de S3. Después de activar las etiquetas de asignación de costos, AWS las utiliza para organizar los costos de los recursos en el informe de asignación de costos. Las etiquetas de asignación de costos solo se pueden utilizar para etiquetar buckets. Para obtener información sobre las etiquetas utilizadas para etiquetación de objetos, consulte Categorización del almacenamiento mediante etiquetas.

El informe mensual de asignación de costes muestra el uso de AWS para la cuenta por categoría de producto y usuario de cuenta vinculada. El informe contiene las mismas partidas que el informe detallado de facturación (consulte Cómo interpretar los informes de facturación y de uso de AWS para Amazon S3) y columnas adicionales para las claves de etiquetas.

AWS proporciona dos tipos de etiquetas de asignación de costos, una etiqueta generada por AWS y las etiquetas definidas por el usuario. AWS define, crea y aplica la etiqueta `createdBy` generada por AWS después de un evento CreateBucket de Amazon S3. Las etiquetas definidas por el usuario son etiquetas que usted define, crea y aplica al bucket de S3.

Debe activar ambos tipos de etiquetas por separado en la consola Billing and Cost Management para que puedan aparecer en los informes de facturación. Para obtener más información acerca de las etiquetas generadas por AWS, consulte Etiquetas de asignación de costos generadas por AWS.

- Para crear etiquetas en la consola, consulte Visualización de las propiedades para un bucket de S3.
- Para crear etiquetas mediante la API de Amazon S3, consulte Etiquetado de bucket PUT en la referencia de API de Amazon Simple Storage Service.
- Para crear etiquetas con la AWS CLI, consulte put-bucket-tagging en la Referencia de comandos de la AWS CLI.
- Para obtener más información acerca de la activación de etiquetas, consulte Uso de etiquetas de asignación de costos en la Guía del usuario de AWS Billing.

Etiquetas de asignación de costos definidas por el usuario

Una etiqueta de asignación de costos definida por el usuario tiene los siguientes componentes:

- La clave de la etiqueta. La clave de la etiqueta es el nombre de la etiqueta. Por ejemplo, en la etiqueta proyecto/Trinity, proyecto es la clave. La clave de la etiqueta es una cadena que distingue entre mayúsculas y minúsculas y que puede contener de 1 a 128 caracteres Unicode.

- El valor de la etiqueta. El valor de la etiqueta es una cadena obligatoria. Por ejemplo, en la etiqueta proyecto/Trinity, Trinity es el valor. El valor de la etiqueta es una cadena que distingue entre mayúsculas y minúsculas y que puede contener de 0 a 256 caracteres Unicode.

Para obtener información detallada acerca de los caracteres permitidos para las etiquetas definidas por el usuario y otras restricciones, consulte Restricciones de las etiquetas definidas por el usuario en la Guía del usuario de AWS Billing. Para obtener más información acerca de las etiquetas definidas por el usuario, consulte Etiquetas de asignación de costos definidas por el usuario en la Guía del usuario de AWS Billing.

Etiquetas de bucket de S3

Cada bucket de S3 tiene un conjunto de etiquetas. Un conjunto de etiquetas contiene todas las etiquetas que están asignadas a ese bucket. Un conjunto de etiquetas puede contener hasta 50 etiquetas, y también puede estar vacío. Las claves deben ser únicas dentro de un conjunto de etiquetas, pero los valores de un conjunto de etiquetas no tienen que ser únicos. Por ejemplo, puede tener el mismo valor en los conjuntos de etiquetas denominados proyecto/Trinity and centro-de-costos/Trinity.

En un bucket, si añade una etiqueta con la misma clave que una etiqueta existente, el valor nuevo sobrescribe el anterior.

AWS no aplica ningún significado semántico a las etiquetas. Las etiquetas se interpretan estrictamente como cadenas de caracteres.

Para agregar, enumerar, editar o eliminar etiquetas, puede utilizar la consola de Amazon S3, AWS Command Line Interface (AWS CLI) o la API de Amazon S3.

Más información

- Uso de etiquetas de asignación de costos en la Guía del usuario de AWS Billing.
- Cómo interpretar los informes de facturación y de uso de AWS para Amazon S3
- Informes de AWS Billing para Amazon S3

Informes de facturación y uso de Amazon S3

> **Important**
>
> El 13 de mayo de 2024, empezamos a implementar un cambio para eliminar los cargos por solicitudes no autorizadas que no haya iniciado el propietario del bucket. Una vez que se complete la implementación de este cambio, los propietarios de los buckets nunca incurrirán en cargos por solicitud o ancho de banda por las solicitudes que devuelvan errores `AccessDenied` (HTTP `403 Forbidden`) cuando estas solicitudes se inicien desde fuera de la cuenta de AWS individual u organización de AWS. Para obtener más información sobre una lista completa de códigos de estado `3XX` y `4XX` HTTP que no se facturarán, consulte Facturación para respuestas de errores de Amazon S3. Este cambio de facturación no requiere actualizaciones en las aplicaciones y se aplica a todos los buckets de S3. Cuando se haya completado la implementación de este cambio en todas las Regiones de AWS, actualizaremos nuestra documentación.

Si se utiliza Amazon S3, no es necesario pagar ninguna cuota inicial ni comprometerse a almacenar una cantidad de contenido determinada. Como con otros Servicios de AWS, pagará por uso y únicamente por lo que use.

AWS proporciona los siguientes informes para Amazon S3:

- Informes de facturación: son varios informes que proporcionan vistas de alto nivel de toda la actividad de los Servicios de AWS que está utilizando, incluido Amazon S3. AWS siempre factura al propietario del bucket de S3 las tarifas de Amazon S3, a menos que el bucket se haya creado como un bucket de pago por solicitante. Para obtener más información acerca del pago por solicitante, consulte Utilización de buckets de pago por solicitante para transferencias de almacenamiento y uso. Para obtener más información acerca de los informes de facturación, consulte Informes de AWS Billing para Amazon S3.
- Informe de uso: es un resumen de la actividad de un servicio específico, agrupado por hora, día o mes. Puede elegir el tipo de uso y la operación que desea incluir. También puede elegir la forma en que se agregan los datos. Para obtener más información, consulte Informe de uso de AWS para Amazon S3.

Los siguientes temas proporcionan información acerca de los informes de facturación y uso de Amazon S3.

Temas

Informes de AWS Billing para Amazon S3

La factura mensual de AWS separa la información de uso y los costos por función y Servicio de AWS. Hay varios informes de AWS Billing disponibles, como el informe mensual, el informe de asignación de costos y los informes detallados de facturación. Para obtener información acerca de cómo ver los informes de facturación, consulte Visualización de su factura en la Guía del usuario de AWS Billing.

Para realizar un seguimiento de AWS y proporcionar los cargos estimados asociados a su cuenta, puede configurar Cost and Usage Reports de AWS. Para obtener más información, consulte What are AWSCost and Usage Reports? en la Guía de exportación de datos de AWS.

También puede descargar un informe de uso que da más detalles sobre su uso del almacenamiento de Amazon S3 que los informes de facturación. Para obtener más información, consulte Informe de uso de AWS para Amazon S3.

En la siguiente tabla se muestran los cargos relacionados con el uso de Amazon S3.

Cargos por uso de Amazon S3

Cargo	Comentarios
Almacenamiento	Se paga por almacenar objetos en los buckets de S3. La tarifa que se le cobra depende del tamaño de los objetos, el tiempo que se almacenan durante el mes y la clase de almacenamiento. Amazon S3 ofrece las siguientes clases de almacenamiento: S3 Standard, S3 Express One Zone, S3 Intelligent-Tiering, S3 Standard-IA (IA para acceso poco frecuente), S3 One Zone-IA, S3 Glacier Instant Retrieval, S3 Glacier Flexible Retrieval

Cargo	Comentarios
	, S3 Glacier Deep Archive o Almacenamiento de redundancia reducida (RRS). Para obtener más información acerca de las clases de almacenamiento, consulte Uso de las clases de almacenamiento de Amazon S3. Tenga en cuenta que si tiene habilitado el control de versiones de S3, se le cobrará por cada versión de un objeto que se retenga. Para obtener más información sobre el control de versiones, consulte Cómo funciona S3 Versioning.
Monitorización y automatización	Se abona una tarifa mensual por monitoreo y automatización por cada objeto almacenado en la clase de almacenamiento S3 Intelligent-Tiering para controlar los patrones de acceso y mover objetos entre las capas de acceso en S3 Intelligent-Tiering.
Solicitudes	Se paga por las solicitudes, por ejemplo, solicitudes `GET`, realizadas en los buckets y los objetos de S3. Esto incluye las solicitudes del ciclo de vida. Las tarifas para las solicitudes dependen del tipo de solicitud que se realiza. Para obtener más información acerca de los precios de las solicitudes, consulte Precios de Amazon S3.
Recuperaciones	Se paga por recuperar los objetos que están almacenados en el almacenamiento S3 Standard-IA, S3 One Zone-IA, S3 Glacier Instant Retrieval, S3 Glacier Flexible Retrieval y S3 Glacier Deep Archive.

Cargo	Comentarios
Eliminaciones anticipadas	Si se elimina un objeto almacenado en el almacenamiento S3 Standard-IA, S3 One Zone-IA, S3 Glacier Instant Retrieval, S3 Glacier Flexible Retrieval o S3 Glacier Deep Archive antes de que se haya cumplido el compromiso de almacenamiento mínimo, se pagará una tarifa de eliminación anticipada para ese objeto.
Administrar el almacenamiento	Paga por las características de administración de almacenamiento (Inventario de Amazon S3, análisis y etiquetado de objetos) que están activadas en los buckets de la cuenta.
Ancho de banda	Paga por el ancho de banda dentro y fuera de Amazon S3, excepto por lo siguiente: • Datos entrantes transferidos desde Internet • Datos salientes transferidos hacia una instancia de Amazon Elastic Compute Cloud (Amazon EC2), cuando la instancia se encuentra en la misma Región de AWS que el bucket de S3 • Transferencia de datos hacia Amazon CloudFront (CloudFront) También deberá abonar una tarifa por las transferencias de datos realizadas mediante Aceleración de transferencias de Amazon S3.

Para obtener información detallada acerca de los cargos por uso de Amazon S3 relativas a almacenamiento, transferencia de datos y servicios, consulte Precios de Amazon S3 y las Preguntas frecuentes sobre Amazon S3.

Para obtener información acerca del significado de los códigos y las abreviaturas utilizados en los informes de facturación y de uso de Amazon S3, consulte Cómo interpretar los informes de facturación y de uso de AWS para Amazon S3.

Más información

- Informe de uso de AWS para Amazon S3
- Uso de etiquetas de buckets de S3 de asignación de costos
- Administración de facturación y costos de AWS Billing
- Precios de Amazon S3

Informe de uso de AWS para Amazon S3

Cuando se descarga un informe de uso, se pueden obtener los datos de uso agregados por hora, día o mes. El informe de uso de Amazon S3 enumera las operaciones según el tipo de uso y la Región de AWS. Para obtener informes más detallados acerca del uso que hace del almacenamiento de Amazon S3, descargue los informes de uso de AWS generados de forma dinámica. Puede elegir el tipo de uso, la operación y el periodo de tiempo que desea incluir. También puede elegir la forma en que se agregan los datos. Para obtener más información acerca de los informes de uso, consulte AWSInforme de uso de en la Guía del usuario de exportación de datos de AWS.

El informe de uso de Amazon S3 incluye la siguiente información:

- Service (Servicio): Amazon S3
- Operation (Operación): la operación realizada en el bucket o en el objeto. Para obtener una explicación detallada de las operaciones de Amazon S3, consulte Seguimiento de las operaciones en los informes de uso.
- UsageType: uno de los siguientes valores:
  - Un código que identifica el tipo de almacenamiento
  - Un código que identifica el tipo de solicitud
  - Un código que identifica el tipo de recuperación
  - Un código que identifica el tipo de transferencia de datos.

  - Un código que identifica las eliminaciones anticipadas del almacenamiento S3 Intelligent-Tiering, S3 Standard-IA, S3 One Zone-Infrequent Access (S3 One Zone-IA), S3 Glacier Flexible Retrieval o S3 Glacier Deep Archive
  - `StorageObjectCount`: el número de objetos almacenados en un bucket determinado

  Para obtener una explicación detallada de los tipos de uso de Amazon S3, consulte Cómo interpretar los informes de facturación y de uso de AWS para Amazon S3.
- Resource (Recurso): el nombre del bucket asociado al uso indicado.
- StartTime (Hora de inicio): la hora de inicio del día en que se realizó el uso, en horario universal coordinado (UTC).
- EndTime (Hora de finalización): la hora de finalización del día en que se realizó el uso, en horario universal coordinado (UTC).
- UsageValue (Valor de uso): uno de los siguientes valores de volumen: La unidad de medida típica para los datos es gigabytes (GB). Sin embargo, según el servicio y del informe, podrían aparecer terabytes (TB).
  - El número de solicitudes durante el periodo especificado.
  - La cantidad de datos transferidos.
  - La cantidad de datos almacenados en una hora determinada
  - La cantidad de datos relacionados con las restauraciones desde el almacenamiento S3 Standard-IA, S3 One Zone-IA, S3 Glacier Flexible Retrieval o S3 Glacier Deep Archive

> **Tip**
>
> Para obtener información detallada acerca de cada solicitud que Amazon S3 recibe de sus objetos, active el registro de acceso del servidor para sus buckets. Para obtener más información, consulte Registro de solicitudes con registro de acceso al servidor.

Puede descargar un informe de uso como un archivo XML o como un archivo de valores separados por comas (CSV). A continuación se muestra un ejemplo de informe de uso con formato CSV abierto en una aplicación de hoja de cálculo.

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

Para obtener más información, consulte Cómo interpretar los informes de facturación y de uso de AWS para Amazon S3.

Descarga del informe de uso de AWS

Puede descargar un informe de uso como archivo XML o CSV.

Para descargar el informe de uso

1. Inicie sesión AWS Management Console Management Console y abra la consola de Amazon S3 en https://console.aws.amazon.com/s3/.
2. En la barra de título, elija su nombre de usuario o ID de cuenta y, a continuación, Administración de facturación y costos.
3. En el panel de navegación, seleccione Informes de uso y costo.
4. En AWS informe de uso, elija Crear un informe de uso.
5. En Descargar el informe de uso, elija las opciones siguientes:
   - Servicios: elija Amazon Simple Storage Service.
   - Usage Types (Tipos de uso): para obtener una explicación detallada de los tipos de uso de Amazon S3, consulte Cómo interpretar los informes de facturación y de uso de AWS para Amazon S3.
   - Operation (Operación): para obtener una explicación detallada de las operaciones de Amazon S3, consulte Seguimiento de las operaciones en los informes de uso.
   - Time Period (Periodo de tiempo): seleccione el periodo de tiempo que desea que abarque el informe.
   - Report Granularity (Grado de detalle de informe): indique si desea que el informe incluya subtotales por hora, por día o por mes.
6. Elija Descargar, el formato de descarga (Informe XML o Informe CSV) y, a continuación, siga las instrucciones para abrir o guardar el informe.

Más información

- Cómo interpretar los informes de facturación y de uso de AWS para Amazon S3
- Informes de AWS Billing para Amazon S3

Cómo interpretar los informes de facturación y de uso de AWS para Amazon S3

> **Important**
>
> El 13 de mayo de 2024, empezamos a implementar un cambio para eliminar los cargos por solicitudes no autorizadas que no haya iniciado el propietario del bucket. Una vez que se complete la implementación de este cambio, los propietarios de los buckets nunca incurrirán en cargos por solicitud o ancho de banda por las solicitudes que devuelvan errores `AccessDenied` (HTTP `403 Forbidden`) cuando estas solicitudes se inicien desde fuera de la cuenta de AWS individual u organización de AWS. Para obtener más información sobre una lista completa de códigos de estado `3XX` y `4XX` HTTP que no se facturarán, consulte Facturación para respuestas de errores de Amazon S3. Este cambio de facturación no requiere actualizaciones en las aplicaciones y se aplica a todos los buckets de S3. Cuando se haya completado la implementación de este cambio en todas las Regiones de AWS, actualizaremos nuestra documentación.

Los informes de facturación y de uso de Amazon S3 utilizan códigos y abreviaturas. Para los tipos de uso de la siguiente tabla, reemplace *`region`*, *`region1`* y *`region2`* por las abreviaturas de esta lista:

- APE1: Asia-Pacífico (Hong Kong)
- APN1: Asia-Pacífico (Tokio)
- APN2: Asia-Pacífico (Seúl)
- APN3: Asia Pacífico (Osaka)
- APS1: Asia-Pacífico (Singapur)
- APS2: Asia-Pacífico (Sídney)
- APS3: Asia-Pacífico (Mumbai)
- APS4: Asia Pacífico (Yakarta)
- APS5: Asia-Pacífico (Hyderabad)

- APS6: Asia-Pacífico (Melbourne)
- CAN1: Canadá (Central)
- CAN2: oeste de Canadá (Calgary)
- CNN1: China (Pekín)
- CNW1: China (Ningxia)
- AFS1: África (Ciudad del Cabo)
- EUC2: Europa (Zúrich)
- EUN1: Europa (Estocolmo)
- EUS2: Europa (España)
- EUC1: Europa (Fráncfort)
- EU: Europa (Irlanda)
- EUS1: Europa (Milán)
- EUW2: Europa (Londres)
- UEW3: Europa (París)
- ILC1: Israel (Tel Aviv)
- MEC1: Oriente Medio (EAU)
- MES1: Medio Oriente (Baréin)
- SAE1: América del Sur (São Paulo)
- UGW1: AWS GovCloud (EE. UU. Oeste)
- UGE1: AWS GovCloud (EE. UU. Este)
- USE1 (o sin prefijo): EE. UU. Este (Norte de Virginia)
- USE2: EE. UU. Este (Ohio)
- USW1: EE. UU. Oeste (Norte de California)
- USW2: EE. UU. Oeste (Oregón)

Para los tipos de uso de puntos de acceso de varias regiones de S3 de la siguiente tabla, reemplace *`regiongroup1`* y *`regiongroup2`* por las abreviaturas de esta lista:

- AP: Asia-Pacífico
- AU: Australia
- EU: Europa

- IN: India
- NA: América del Norte
- SA: América del Sur

Los grupos de regiones son agrupaciones geográficas de varias Regiones de AWS. Para obtener más información, consulte Regiones y zonas de disponibilidad. Para obtener información acerca de los precios según Región de AWS, consulte Precios de Amazon S3.

En la primera columna de la siguiente tabla se enumeran los tipos de uso que aparecen en los informes de facturación y de uso. La unidad de medida típica para los datos es gigabytes (GB). Sin embargo, según el servicio y del informe, podrían aparecer terabytes (TB).

Tipos de uso

Tipo de uso	Unidades	Grado de detalle	Descripción
region1-*region2*-AWS-In-A Bytes	GB	Por hora	Es la cantidad de datos acelerados transferidos a la *region1* desde la *region2*.
region1-*region2*-AWS-In-A Bytes-T1	GB	Por hora	Es la cantidad de datos acelerados de T1 transferidos a la *region1* desde la *region2*, donde T1 se refiere a las solicitudes de CloudFront a puntos de presencia (POPs) en Estados Unidos, Europa y Japón.
region1-*region2*-AWS-In-A Bytes-T2	GB	Por hora	Es la cantidad de datos acelerados de T2 transferidos a la *region1* desde la *region2*, donde T2 se

Tipo de uso	Unidades	Grado de detalle	Descripción
			refiere a las solicitudes de CloudFront a POPs en el resto de ubicaciones periféricas de AWS.
region1-*region2*-AWS-In-Bytes	GB	Por hora	Es la cantidad de datos transferidos a la *region1* desde la *region2*.
region1-*region2*-AWS-Out-ABytes	GB	Por hora	Es la cantidad de datos acelerados transferidos desde la *region1* a la *region2*.
region1-*region2*-AWS-Out-ABytes-T1	GB	Por hora	La cantidad de datos acelerados de T1 transferidos desde la *region1* a la *region2*, donde T1 se refiere a las solicitudes de CloudFront a POP en Estados Unidos, Europa y Japón
region1-*region2*-AWS-Out-ABytes-T2	GB	Por hora	Es la cantidad de datos acelerados de T2 transferidos desde la *region1* a la *region2*, donde T2 se refiere a las solicitudes de CloudFron t a puntos de presencia (POP) en el resto de ubicaciones periféricas de AWS.

Tipo de uso	Unidades	Grado de detalle	Descripción
region1-*region2*-AWS-Out-Bytes	GB	Por hora	Es la cantidad de datos transferidos desde la *region1* a la *region2*.
region-BatchOperations-Jobs	Recuento	Por hora	El número de trabajos de operaciones por lotes de S3 realizados
region-BatchOperations-Objects	Recuento	Por hora	El número de operaciones de objeto realizadas por operaciones por lotes de S3
region-Bulk-Retrieval-Bytes	GB	Por hora	La cantidad de datos recuperados mediante solicitudes S3 Glacier Flexible Retrieval o S3 Glacier Deep Archive en bloque
region-BytesDeleted-GDA	GB	Mensual	Es la cantidad de datos eliminados por una operación DeleteObject del almacenamiento S3 Glacier Deep Archive.
region-BytesDeleted-GIR	GB	Mensual	Es la cantidad de datos eliminados por una operación DeleteObject del almacenamiento S3 Glacier Instant Retrieval.

Tipo de uso	Unidades	Grado de detalle	Descripción
region-BytesDeleted-GLACIER	GB	Mensual	Es la cantidad de datos eliminados por una operación DeleteObject del almacenamiento S3 Glacier Flexible Retrieval.
region-BytesDeleted-INT	GB	Mensual	Es la cantidad de datos eliminados por una operación DeleteObject del almacenamiento S3 Intelligent-Tiering.
region-BytesDeleted-RRS	GB	Mensual	Es la cantidad de datos eliminados por una operación DeleteObject del Almacenamiento de redundancia reducida (RRS).
region-BytesDeleted-SIA	GB	Mensual	Es la cantidad de datos eliminados por una operación DeleteObject del almacenamiento S3 Standard-IA.
region-BytesDeleted-STANDARD	GB	Mensual	Es la cantidad de datos eliminados por una operación DeleteObject del almacenamiento S3 Standard.

Tipo de uso	Unidades	Grado de detalle	Descripción
region-BytesDeleted-ZIA	GB	Mensual	Es la cantidad de datos eliminados por una operación DeleteObject del almacenamiento S3 One Zone-IA.
region-C3DataTransfer-In-Bytes	GB	Por hora	La cantidad de datos transferidos a Amazon S3 de Amazon EC2 dentro de la misma Región de AWS
region-C3DataTransfer-Out-Bytes	GB	Por hora	La cantidad de datos transferidos de Amazon S3 a Amazon EC2 dentro de la misma Región de AWS
region-CloudFront-In-Bytes	GB	Por hora	La cantidad de datos transferidos a una Región de AWS de una distribución de CloudFront
region-CloudFront-Out-Bytes	GB	Por hora	La cantidad de datos transferidos de una Región de AWS a una distribución de CloudFront
region-DataTransfer-In-Bytes	GB	Por hora	La cantidad de datos transferidos a Amazon S3 desde Internet

Tipo de uso	Unidades	Grado de detalle	Descripción
region-DataTransfer-Out-Bytes	GB	Por hora	La cantidad de datos transferidos desde Amazon S3 a Internet[1]
region-DataTransfer-Regional-Bytes	GB	Por hora	La cantidad de datos transferidos de Amazon S3 a los recursos de AWS dentro de la misma Región de AWS
region-EarlyDelete-ByteHrs	GB-horas	Por hora	Uso de almacenamiento prorrateado para los objetos eliminados del almacenamiento S3 Glacier Flexible Retrieval antes de que finalice el compromiso mínimo de 90 días[2]
region-EarlyDelete-GDA	GB-horas	Por hora	Uso de almacenamiento prorrateado para los objetos eliminados del almacenamiento S3 Glacier Deep Archive antes de que finalice el compromiso mínimo de 180 días [2]

Tipo de uso	Unidades	Grado de detalle	Descripción
region-EarlyDelete-GIR	GB-horas	Por hora	Uso de almacenamiento prorrateado para los objetos eliminados del almacenamiento S3 Glacier Instant Retrieval antes de que finalice el compromiso mínimo de 90 días.
region-EarlyDelete-GIR-SmObjects	GB-horas	Por hora	Uso de almacenamiento prorrateado para los objetos pequeños (de menos de 128 KB) que se eliminaron del almacenamiento S3 Glacier Instant Retrieval antes de que finalizara el compromiso mínimo de 90 días.
region-EarlyDelete-SIA	GB-horas	Por hora	Uso de almacenamiento prorrateado para los objetos eliminados del almacenamiento S3 Standard-IA antes de que finalice el compromiso mínimo de 30 días[3]

Tipo de uso	Unidades	Grado de detalle	Descripción
region-EarlyDelete-SIA-SmObjects	GB-horas	Por hora	Uso de almacenamiento prorrateado para los objetos pequeños (de menos de 128 KB) que se eliminaron del almacenamiento S3 Standard-IA antes de que finalizara el compromiso mínimo de 30 días[3]
region-EarlyDelete-ZIA	GB-horas	Por hora	Uso de almacenamiento prorrateado para los objetos eliminados del almacenamiento S3 One Zone-IA antes de que finalice el compromiso mínimo de 30 días[3]
region-EarlyDelete-ZIA-SmObjects	GB-horas	Por hora	Uso de almacenamiento prorrateado para los objetos pequeños (de menos de 128 KB) que se eliminaron del almacenamiento S3 One Zone-IA antes de que finalizara el compromiso mínimo de 30 días[3]
region-Expedited-Retrieval-Bytes	GB	Por hora	La cantidad de datos recuperados con solicitudes rápidas de S3 Glacier Flexible Retrieval

Tipo de uso	Unidades	Grado de detalle	Descripción
region-Inventory-Objects Listed	Objects	Por hora	El número de objetos enumerados para un grupo de objetos (los objetos se agrupan por bucket o por prefijo) con una lista de inventario
region-Monitoring-Automation-INT	Objetos	Por hora	El número de objetos únicos monitorizados y designados automáticamente por capas en la clase de almacenamiento S3 Intelligent-Tiering
region-MRAP-Out-Bytes	GB	Por hora	Es la cantidad de datos transferidos a través de un punto de conexión de puntos de acceso de varias regiones de S3 fuera de los buckets de una región (precio de enrutamiento de datos de MRAP).
region-MRAP-In-Bytes	GB	Por hora	Es la cantidad de datos transferidos a través de un punto de conexión de puntos de acceso de varias regiones de S3 fuera de los buckets de una región (precio de enrutamiento de datos de MRAP).

Tipo de uso	Unidades	Grado de detalle	Descripción
regiongroup1-regiongroup2--MRAP-Out-Bytes	GB	Por hora	Es la cantidad de datos transferidos a través de un punto de conexión de puntos de acceso de varias regiones de S3 desde un bucket del *regiongroup1* a un cliente del *regiongroup2* que se encuentra fuera de la red de AWS.
regiongroup1-regiongroup2--MRAP-In-Bytes	GB	Por hora	Es la cantidad de datos transferidos a través de un punto de conexión de puntos de acceso de varias regiones de S3 a un bucket del *regiongroup1* desde un cliente del *regiongroup2* que se encuentra fuera de la red de AWS.
region-OverwriteBytes-Copy-GDA	GB	Mensual	Es la cantidad de datos anulados por una operación CopyObject del almacenamiento S3 Glacier Deep Archive.
region-OverwriteBytes-Copy-GIR	GB	Mensual	Es la cantidad de datos anulados por una operación CopyObject del almacenamiento S3 Glacier Instant Retrieval.

Tipo de uso	Unidades	Grado de detalle	Descripción
region-OverwriteBytes-Copy-GLACIER	GB	Mensual	Es la cantidad de datos anulados por una operación CopyObject del almacenamiento S3 Glacier Flexible Retrieval.
region-OverwriteBytes-Copy-INT	GB	Mensual	Es la cantidad de datos anulados por una operación CopyObject del almacenamiento S3 Intelligent-Tiering.
region-OverwriteBytes-Copy-RRS	GB	Mensual	Es la cantidad de datos anulados por una operación CopyObject del Almacenamiento de redundancia reducida (RRS).
region-OverwriteBytes-Copy-SIA	GB	Mensual	Es la cantidad de datos anulados por una operación CopyObject del almacenamiento S3 Standard-IA.
region-OverwriteBytes-Copy-STANDARD	GB	Mensual	Es la cantidad de datos anulados por una operación CopyObject del almacenamiento S3 Standard.

Tipo de uso	Unidades	Grado de detalle	Descripción
region-OverwriteBytes-Copy-ZIA	GB	Mensual	Es la cantidad de datos anulados por una operación CopyObject del almacenamiento S3 One Zone-IA.
region-OverwriteBytes-Put-GDA	GB	Mensual	Es la cantidad de datos anulados por una operación PutObject del almacenamiento S3 Glacier Deep Archive.
region-OverwriteBytes-Put-GIR	GB	Mensual	Es la cantidad de datos anulados por una operación PutObject del almacenamiento S3 Glacier Instant Retrieval.
region-OverwriteBytes-Put-GLACIER	GB	Mensual	Es la cantidad de datos anulados por una operación PutObject del almacenamiento S3 Glacier Flexible Retrieval.
region-OverwriteBytes-Put-INT	GB	Mensual	Es la cantidad de datos anulados por una operación PutObject del almacenamiento S3 Intelligent-Tiering.

Tipo de uso	Unidades	Grado de detalle	Descripción
region-OverwriteBytes-Put-RRS	GB	Mensual	Es la cantidad de datos anulados por una operación PutObject del Almacenamiento de redundancia reducida (RRS).
region-OverwriteBytes-Put-SIA	GB	Mensual	Es la cantidad de datos anulados por una operación PutObject del almacenamiento S3 Standard-IA.
region-OverwriteBytes-Put-STANDARD	GB	Mensual	Es la cantidad de datos anulados por una operación PutObject del almacenamiento S3 Standard.
region-OverwriteBytes-Put-ZIA	GB	Mensual	Es la cantidad de datos anulados por una operación PutObject del almacenamiento S3 One Zone-IA.

Tipo de uso	Unidades	Grado de detalle	Descripción
region1-*region2*-S3RTC-In-Bytes	GB	Mensual	La cantidad de datos transferidos para el Control del tiempo de replicación de S3 (S3 RTC) desde *region2* a *region1* mediante las operaciones PutObjectReplTime , GetObjectReplTime , InitiateMultipartUploadReplTime , UploadPartReplTime , CompleteMultipartUploadReplTime y WriteACLReplTime
region1-*region2*-S3RTC-Out-Bytes	GB	Mensual	La cantidad de datos transferidos para el Control del tiempo de replicación de S3 (S3 RTC) desde *region1* a *region2* mediante las operaciones PutObjectReplTime , GetObjectReplTime , InitiateMultipartUploadReplTime , UploadPartReplTime , CompleteMultipartUploadReplTime y WriteACLReplTime

Tipo de uso	Unidades	Grado de detalle	Descripción
region-Requests-GDA-Tier1	Recuento	Por hora	Es el número de solicitudes PUT, COPY, POST, CreateMultipartUpload , UploadPart o CompleteMultipartUpload en objetos S3 Glacier Deep Archive[6].
region-Requests-GDA-Tier2	Recuento	Por hora	Es el número de solicitudes GET y HEAD en objetos S3 Glacier Deep Archive.
region-Requests-GDA-Tier3	Recuento	Por hora	El número de solicitudes de restauración estándar de S3 Glacier Deep Archive
region-Requests-GDA-Tier5	Recuento	Por hora	El número de solicitudes de restauración masiva de S3 Glacier Deep Archive
region-Requests-GIR-Tier1	Recuento	Por hora	Es el número de solicitudes PUT, COPY o POST realizadas con objetos S3 Glacier Instant Retrieval.
region-Requests-GIR-Tier2	Recuento	Por hora	Es el número de solicitudes GET y todas las demás solicitudes que no son S3 Glacier Instant Retrieval -Tier1 con objetos S3 Glacier Instant Retrieval.

Tipo de uso	Unidades	Grado de detalle	Descripción
region-Requests-GLACIER-Tier1	Recuento	Por hora	Es el número de solicitudes PUT, COPY, POST, CreateMultipartUpload , UploadPart o CompleteMultipartUpload realizadas con objetos S3 Glacier Flexible Retrieval[6].
region-Requests-GLACIER-Tier2	Recuento	Por hora	Es el número de solicitudes GET y todas las demás solicitudes no enumeradas con objetos S3 Glacier Flexible Retrieval.
region-Requests-INT-Tier1	Recuento	Por hora	Es el número de solicitudes PUT, COPY o POST realizadas con objetos S3 Intelligent-Tiering.
region-Requests-INT-Tier2	Recuento	Por hora	Es el número de solicitudes GET y el resto de solicitudes que no son Tier1 realizadas para objetos S3 Intelligent-Tiering.
region-Requests-SIA-Tier1	Recuento	Por hora	Es el número de solicitudes PUT, COPY o POST realizadas con objetos S3 Standard-IA.

Tipo de uso	Unidades	Grado de detalle	Descripción
region-Requests-SIA-Tier2	Recuento	Por hora	Es el número de solicitud es GET y todas las demás que no son S3 Glacier Instant Retrieval-Tier1 realizadas con objetos S3 Standard-IA.
region-Requests-Tier1	Recuento	Por hora	Es el número de solicitud es PUT, COPY o POST realizadas para S3 Standard, RRS y etiquetas, además de las solicitudes LIST para todos los buckets y objetos.
region-Requests-Tier2	Recuento	Por hora	Es el número de solicitud es GET y todas las demás que no son Tier1.
region-Requests-Tier3	Recuento	Por hora	El número de solicitud es del ciclo de vida a S3 Glacier Flexible Retrieval o S3 Glacier Deep Archive y solicitudes de restauración estándar de S3 Glacier Flexible Retrieval

Tipo de uso	Unidades	Grado de detalle	Descripción
region-Requests-Tier4	Recuento	Por hora	El número de transiciones del ciclo de vida a almacenamiento S3 Glacier Instant Retrieval, S3 Intelligent-Tiering, S3 Standard-IA, o S3 One Zone-IA
region-Requests-Tier5	Recuento	Por hora	El número de solicitudes de restauración S3 Glacier Flexible Retrieval en bloque
region-Requests-Tier6	Recuento	Por hora	El número de solicitudes de restauración S3 Glacier Flexible Retrieval rápidas
region-Requests-Tier8	Recuento	Por hora	Es el número de solicitudes de S3 Access Grants.
region-Requests-XZ-Tier1	Recuento	Por hora	Es el número de solicitudes PUT o COPY realizadas en objetos S3 Express One Zone.
region-Requests-XZ-Tier2	Recuento	Por hora	Es el número de solicitudes GET y todas las demás solicitudes que no son S3 Express One Zone-Tier1 realizadas en objetos S3 Express One Zone.

Tipo de uso	Unidades	Grado de detalle	Descripción
region-Requests-ZIA-Tier1	Recuento	Por hora	Es el número de solicitud es PUT, COPY o POST realizadas en objetos S3 One Zone-IA.
region-Requests-ZIA-Tier2	Recuento	Por hora	Es el número de solicitud es GET y todas las demás solicitudes que no son S3 One Zone-IA-Tier1 realizadas en objetos S3 One Zone-IA.
region-Retrieval-GIR	GB	Por hora	La cantidad de datos recuperados del almacenamiento de S3 Glacier Instant Retrieval.
region-Retrieval-SIA	GB	Por hora	La cantidad de datos recuperados del almacenamiento S3 Standard-IA
region-Retrieval-XZ	GB	Por hora	Es la parte de los datos que supera los 512 KB en una solicitud de recuperación determina da (PUT o COPY) con el almacenamiento S3 Express One Zone.
region-Retrieval-ZIA	GB	Por hora	La cantidad de datos recuperados del almacenamiento S3 One Zone-IA

Tipo de uso	Unidades	Grado de detalle	Descripción
region-S3DSSE-In-Bytes	GB	Mensual	Es la cantidad de datos de doble cifrado de Amazon S3.
region-S3DSSE-Out-Bytes	GB	Mensual	Es la cantidad de datos de doble cifrado descifrados por Amazon S3.
region-S3G-DataTransfer-In-Bytes	GB	Por hora	La cantidad de datos transferidos a Amazon S3 para restaurar objetos desde el almacenamiento de S3 Glacier Flexible Retrieval o S3 Glacier Deep Archive
region-S3G-DataTransfer-Out-Bytes	GB	Por hora	La cantidad de datos transferidos desde Amazon S3 a objetos de transición al almacenamiento de S3 Glacier Flexible Retrieval o S3 Glacier Deep Archive
region-Select-Returned-Bytes	GB	Por hora	La cantidad de datos devueltos con las solicitudes de selección del almacenamiento S3 Standard
region-Select-Returned-GIR-Bytes	GB	Por hora	La cantidad de datos devueltos con solicitudes Select del almacenamiento de S3 Glacier Instant Retrieval.

Tipo de uso	Unidades	Grado de detalle	Descripción
region-Select-Returned-INT-Bytes	GB	Por hora	La cantidad de datos devueltos con las solicitudes de selección del almacenamiento S3 Intelligent-Tiering
region-Select-Returned-SIA-Bytes	GB	Por hora	La cantidad de datos devueltos con las solicitudes de selección del almacenamiento S3 Standard-IA
region-Select-Returned-ZIA-Bytes	GB	Por hora	La cantidad de datos devueltos con las solicitudes de selección del almacenamiento S3 One Zone-IA
region-Select-Scanned-Bytes	GB	Por hora	La cantidad de datos escaneados con las solicitudes de selección del almacenamiento S3 Standard
region-Select-Scanned-GIR-Bytes	GB	Por hora	La cantidad de datos escaneados con solicitudes Select del almacenamiento S3 Glacier Instant Retrieval.

Tipo de uso	Unidades	Grado de detalle	Descripción
region-Select-Scanned-INT-Bytes	GB	Por hora	La cantidad de datos escaneados con las solicitudes de selección del almacenamiento de almacenamiento S3 Intelligent-Tiering
region-Select-Scanned-SIA-Bytes	GB	Por hora	La cantidad de datos escaneados con las solicitudes de selección del almacenamiento S3 Standard-IA
region-Select-Scanned-ZIA-Bytes	GB	Por hora	La cantidad de datos escaneados con las solicitudes de selección de almacenamiento S3 One Zone-IA
region-Standard-Retrieval-Bytes	GB	Por hora	La cantidad de datos recuperados con solicitud es estándar S3 Glacier Flexible Retrieval o S3 Glacier Deep Archive
region-StorageAnalytics-ObjCount	Objects	Por hora	Número de objetos únicos supervisados en cada configuración de análisis de clase de almacenamiento.

Tipo de uso	Unidades	Grado de detalle	Descripción
region-StorageLens-ObjCount	Objects	Por día	El número de objetos únicos de cada panel de S3 Storage Lens del que se realiza un seguimiento mediante métricas y recomenda ciones avanzadas de S3 Storage Lens.
region-StorageLensFreeTier-ObjCount	Objects	Por día	El número de objetos únicos de cada panel de S3 Storage Lens del que se realiza un seguimiento mediante métricas de uso de S3 Storage Lens.
StorageObjectCount	Recuento	Por día	El número de objetos almacenados en un bucket determinado
region-TagStorage-TagHrs	Etiquetas-Horas	Por día	El total de etiquetas de todos los objetos del bucket por hora
region-TimedStorage-ByteHrs	GB-mes	Por día	El número de GB-meses que los datos estuviero n almacenados en el almacenamiento S3 Standard.
region-TimedStorage-GDA-ByteHrs	GB-mes	Por día	El número de GB-meses que los datos estuviero n almacenados en el almacenamiento S3 Glacier Deep Archive

Tipo de uso	Unidades	Grado de detalle	Descripción
region-TimedStorage-GDA-Staging	GB-mes	Por día	El número de GB-meses que los datos estuvieron almacenados en el almacenamiento provisional S3 Glacier Deep Archive
region-TimedStorage-GIR-ByteHrs	GB-mes	Por día	El número de GB-meses que los datos estuvieron almacenados en el almacenamiento S3 Glacier Instant Retrieval
region-TimedStorage-GIR-SmObjects	GB-mes	Por día	El número de GB-meses que los objetos pequeños (de menos de 128 KB) estuvieron almacenados en el almacenamiento S3 Glacier Instant Retrieval
region-TimedStorage-GlacierByteHrs	GB-mes	Por día	El número de GB-meses que los datos estuvieron almacenados en el almacenamiento S3 Glacier Flexible Retrieval
region-TimedStorage-GlacierStaging	GB-mes	Por día	El número de GB-meses que los datos estuvieron almacenados en el almacenamiento provisional S3 Glacier Flexible Retrieval

Tipo de uso	Unidades	Grado de detalle	Descripción
region-TimedStorage-INT-FA-ByteHrs	GB-mes	Por día	El número de GB-meses que los datos estuvieron almacenados en el nivel Frequent Access del almacenamiento S3 Intelligent-Tiering[5]
region-TimedStorage-INT-IA-ByteHrs	GB-mes	Por día	El número de GB-meses que los datos estuvieron almacenados en el nivel Infrequent Access del almacenamiento S3 Intelligent-Tiering
region-TimedStorage-INT-AA-ByteHrs	GB-mes	Por día	El número de GB-meses que los datos estuvieron almacenados en el nivel Archive Access del almacenamiento S3 Intelligent-Tiering
region-TimedStorage-INT-AIA-ByteHrs	GB-mes	Por día	El número de GB-meses que los datos estuvieron almacenados en el nivel Archive Instant Access del almacenamiento S3 Intelligent-Tiering
region-TimedStorage-INT-DAA-ByteHrs	GB-mes	Por día	El número de GB-meses que los datos estuvieron almacenados en el nivel Deep Archive Access del almacenamiento S3 Intelligent-Tiering

Tipo de uso	Unidades	Grado de detalle	Descripción
region-TimedStorage-RRS-ByteHrs	GB-mes	Por día	El número de GB-meses que los datos estuvieron almacenados en el almacenamiento Reduced Redundancy Storage (RRS)
region-TimedStorage-SIA-ByteHrs	GB-mes	Por día	El número de GB-meses que los datos estuvieron almacenados en el almacenamiento S3 Standard-IA
region-TimedStorage-SIA-SmObjects	GB-mes	Por día	El número de GB-meses que los objetos pequeños (de menos de 128 KB) estuvieron almacenados en el almacenamiento S3 Standard-IA[4]
region-TimedStorage-XZ-ByteHrs	GB-mes	Por día	El número de GB-meses que los datos estuvieron almacenados en el almacenamiento S3 Express One Zone
region-TimedStorage-ZIA-ByteHrs	GB-mes	Por día	El número de GB-meses que los datos estuvieron almacenados en el almacenamiento S3 One Zone-IA

Tipo de uso	Unidades	Grado de detalle	Descripción
region-TimedStorage-ZIA-SmObjects	GB-mes	Por día	El número de GB-meses que los objetos pequeños (de menos de 128 KB) estuvieron almacenados en el almacenamiento S3 One Zone-IA
region-Upload-XZ	GB	Por hora	Es la cantidad de datos que supera los 512 KB en una solicitud de carga determinada (PUT o COPY) con S3 Express One Zone.

Notas

1. Si termina una transferencia antes de su finalización, la cantidad de datos que se transfieren podrían superar la cada de datos que recibe la aplicación. Esta discrepancia podría producirse porque no se puede ejecutar instantáneamente una solicitud de terminación de transferencia y una determinada cantidad de datos podría estar en tránsito, pendiente de ejecución de la solicitud de terminación. Estos datos en tránsito se facturan como datos transferidos "fuera".
2. Cuando los objetos que se archivan en la clase de almacenamiento S3 Glacier Instant Retrieval, S3 Glacier Flexible Retrieval o S3 Glacier Deep Archive se eliminan, se anulan o se trasladan a otra clase de almacenamiento antes de que se haya cumplido el compromiso de almacenamiento mínimo, que es de 90 días en el caso de S3 Glacier Instant Retrieval y S3 Glacier Flexible Retrieval, o de 180 días en S3 Glacier Deep Archive, hay un cargo prorrateado por gigabyte para los días restantes.
3. Para objetos que se guardan en el almacenamiento S3 Standard-IA o S3 One Zone-IA, cuando se eliminan, anulan o pasan a otra clase de almacenamiento antes de 30 días, hay un cargo prorrateado por gigabyte para los días restantes.
4. Para objetos pequeños (de menos de 128 KB) que se guardan en el almacenamiento S3 Standard-IA o S3 One Zone-IA, cuando se eliminan, anulan o pasan a otra clase de almacenamiento antes de 30 días, hay un cargo prorrateado por gigabyte para los días restantes.

5. No hay un tamaño mínimo de objeto facturable para los objetos de la clase de almacenamiento S3 Intelligent-Tiering. Los objetos que son más pequeños que 128 KB no se controlan ni son aptos para la organización automática en niveles. Los objetos más pequeños siempre se almacenan en el nivel de acceso frecuente de S3 Intelligent-Tiering.
6. Al iniciar una solicitud `CreateMultipartUpload`, `UploadPart` o `UploadPartCopy` a las clases de almacenamiento S3 Glacier Flexible Retrieval o S3 Glacier Deep Archive, las solicitudes se facturan según las tarifas de las solicitudes S3 Standard hasta que complete la carga multiparte. Una vez finalizada la carga, la solicitud `CompleteMultipartUpload` individual se factura según la tarifa PUT del almacenamiento S3 Glacier de destino. Las partes de carga multiparte en curso para PUT en la clase de almacenamiento S3 Glacier Flexible Retrieval se facturan como S3 Glacier Flexible Retrieval Staging Storage a las tarifas de almacenamiento de S3 Standard hasta que se complete la carga. Del mismo modo, las partes de carga multiparte en curso para PUT en la clase de almacenamiento S3 Glacier Deep Archive se facturan como S3 Glacier Deep Archive Staging Storage a las tarifas de almacenamiento de S3 Standard hasta que se complete la carga.
7. S3 Express One Zone aplica un cargo fijo por solicitud para solicitudes de hasta 512 KB. Se aplica un cargo adicional por GB a las solicitudes PUT y `GET` para la parte de la solicitud superior a 512 KB.
8. Para obtener información sobre las características compatibles con la clase de almacenamiento S3 Express One Zone, consulte Características de Amazon S3 no compatibles con S3 Express One Zone.
9. Los tipos de uso con unidades que se facturan en GB se calculan en bytes en los informes de uso.
10. Un GB-mes se obtiene tomando el número total de GB-horas, sumándolo en el transcurso de un mes y dividiéndolo por el número de horas de ese mes. Para obtener más información, consulte Preguntas frecuentes: ¿Cómo se me cobrará y facturará por el uso de Amazon S3?

> **Note**
>
> En general, a los propietarios de los buckets de S3 se les facturan las solicitudes con respuestas correctas HTTP `200 OK` y respuestas de error del cliente HTTP `4XX`. A los propietarios de los buckets no se les facturan las respuestas a errores del servidor HTTP `5XX`, como los errores HTTP `503 Slow Down`. Para obtener más información sobre los códigos de error de S3 en HTTP `3XX` y los códigos de estado `4XX` que no se facturan, consulte Facturación para respuestas de errores de Amazon S3. Para obtener más

información sobre los cargos de facturación si el bucket está configurado como un bucket de pago por solicitante, consulte Cómo funcionan los pagos por solicitante.

Seguimiento de las operaciones en los informes de uso

Las operaciones describen la acción realizada con un objeto o un bucket de AWS por el tipo de uso especificado. Las operaciones se indican mediante códigos que expresan claramente su significado, como `PutObject` o `ListBucket`. Para ver qué acciones realizadas en el bucket generaron un tipo de uso específico, utilice estos códigos. Cuando cree un informe de uso, puede seleccionar All Operations (Todas las operaciones) para que incluya todas las operaciones, o una operación específica, como, por ejemplo, `GetObject`.

Más información

- Informe de uso de AWS para Amazon S3
- Informes de AWS Billing para Amazon S3
- Precios de Amazon S3
- Preguntas frecuentes sobre Amazon S3

Facturación para respuestas de errores de Amazon S3

Important

El 13 de mayo de 2024, empezamos a implementar un cambio para eliminar los cargos por solicitudes no autorizadas que no haya iniciado el propietario del bucket. Una vez que se complete la implementación de este cambio, los propietarios de los buckets nunca incurrirán en cargos por solicitud o ancho de banda por las solicitudes que devuelvan errores `AccessDenied` (HTTP `403 Forbidden`) cuando estas solicitudes se inicien desde fuera de la cuenta de AWS individual u organización de AWS. La página actual muestra una lista completa de los códigos de estado HTTP `3XX` y `4XX` que no se facturarán. Este cambio de facturación no requiere actualizaciones en las aplicaciones y se aplica a todos los buckets de S3. Cuando se haya completado la implementación de este cambio en todas las Regiones de AWS, actualizaremos nuestra documentación.

En general, a los propietarios de los buckets de S3 se les facturan las solicitudes con respuestas correctas HTTP `200 OK` y respuestas de error del cliente HTTP `4XX`. A los propietarios de los buckets no se les facturan las respuestas a errores del servidor HTTP `5XX`, como los errores HTTP `503 Slow Down`. Para obtener más información sobre los cargos de facturación si el bucket está configurado como un bucket de pago por solicitante, consulte Cómo funcionan los pagos por solicitante.

En la siguiente tabla, se muestran los códigos de error específicos en códigos de estado HTTP `3XX` y HTTP `4XX` que no se facturan. Para buckets configurados con el alojamiento de sitios web, se seguirán cobrando los cargos de solicitud y de otro tipo cuando S3 devuelva un documento de error personalizado o cuando se trate de redireccionamientos personalizados.

> **Note**
>
> Para `AccessDenied` (HTTP `403 Forbidden`), S3 no cobra al propietario del bucket cuando la solicitud se inicia fuera de la cuenta de AWS individual del propietario del bucket o de la organización de AWS del propietario del bucket.

Código de estado HTTP	Código de error	Descripción del código de error	
301 Moved Permanently (Desplazado permanentemente)	PermanentRedirect	El bucket al que intenta acceder se debe direccionar mediante el punto de conexión especificado. Envíe todas las solicitudes futuras a este punto de conexión.	
	PermanentRedirectControlError	La operación de la API a la que intenta acceder se debe direccionar mediante el punto de conexión especificado. Envíe todas las solicitud	

Código de estado HTTP	Código de error	Descripción del código de error
		es futuras a este punto de conexión.
Redirección temporal 307	TemporaryRedirect	Se le redirige al bucket mientras el servidor del sistema de nombres de dominio (DNS) se está actualizando.
400: solicitud maligna	AuthorizationHeaderMalformed	El encabezado de autorización que ha proporcionado no es válido.
	AuthorizationQueryParametersError	Los parámetros de consulta de autorización que ha proporcionado no son válidos.
	ExpiredToken	El token que ha proporcionado ha caducado.
	IllegalLocationConstraintException	Está intentando acceder a un bucket desde una región diferente a la de donde se encuentra el bucket. Para evitar este error, use la opción `--region`. Por ejemplo: `aws s3 cp awsexample.txt s3://amzn-s3-demo-bucket/ --region ap-east-1 .`

Código de estado HTTP	Código de error	Descripción del código de error	
	InvalidArgument	Este error puede producirse por las siguientes razones: • El argumento especificado no era válido. • A la solicitud le faltaba un encabezado obligatorio. • El argumento especificado estaba incompleto o tenía un formato incorrecto. • El argumento especificado debe tener una longitud superior o igual a 3.	
	InvalidDigest	El valor de Content-MD5 o de la suma de comprobación que especificó no es válido.	
	InvalidEncryptionAlgorithmError	La solicitud de cifrado que especificó no es válida. El valor válido es `AES256`.	

Código de estado HTTP	Código de error	Descripción del código de error	
	InvalidRequest	Este error puede producirse por las siguientes razones: • La solicitud utiliza la versión de firma incorrecta. Use `AWS4-HMAC-SHA256` (Signature Version 4). • Solo se puede crear un punto de acceso para un bucket existente. • El punto de acceso no está en un estado en el que se pueda eliminar. • Solo se puede mostrar un punto de acceso para un bucket existente. • El token siguiente no es válido. • Se debe especificar al menos una acción en una regla del ciclo de vida. •	

Código de estado HTTP	Código de error	Descripción del código de error	
		Se debe especificar al menos una regla del ciclo de vida. • El número de reglas del ciclo de vida no debe superar el límite permitido de 1000 reglas. • El rango para el parámetro `MaxResults` no es válido. • Las solicitudes SOAP deben realizarse a través de una conexión HTTPS. • Los buckets no admiten la aceleración de transferencias de Amazon S3 con nombres no compatibles con DNS. • Los buckets no admiten la aceleración de transferencias de Amazon S3 con puntos (.) en los nombres. •	

Código de estado HTTP	Código de error	Descripción del código de error	
		El punto de conexión de aceleración de transferencia de Amazon S3 solo admite solicitudes de estilo virtual. • La aceleración de transferencias de Amazon S3 no está configurada en este bucket. • La aceleración de transferencias de Amazon S3 está desactivada en este bucket. • La aceleración de transferencias de Amazon S3 no se admite en este bucket. Para obtener ayuda, póngase en contacto con AWS Support. • La aceleración de transferencias de Amazon S3 no se puede habilitar en este bucket. Para obtener ayuda, póngase en	

Código de estado HTTP	Código de error	Descripción del código de error	
		contacto con AWS Support. • Se proporcionan valores contradictorios en los encabezados HTTP y los parámetros de consulta. • Se proporcionan valores contradictorios en los encabezados HTTP y en los campos del formulario POST. • Solicitud CopyObject realizada en objetos de más de 5 GB de tamaño.	
	InvalidSOAPRequest	El cuerpo de la solicitud SOAP no es válido.	
	InvalidStorageClass	La clase de almacenamiento que especificó no es válida.	

Código de estado HTTP	Código de error	Descripción del código de error	
	InvalidTag	La solicitud contiene una entrada de etiqueta que no es válida. Por ejemplo, es posible que la solicitud contenga claves duplicadas, claves o valores demasiado largos o etiquetas de sistema.	
	InvalidToken	El token proporcionado es incorrecto o no es válido de otro modo.	
	InvalidURI	El URI especificado no se pudo analizar.	
	KeyTooLongError	La clave es demasiado larga.	
	MalformedACLError	La ACL proporcionada no tenía un formato válido o no pudo validarse con nuestro esquema publicado.	
	MalformedPOSTRequest	El cuerpo de la solicitud POST no contiene datos multipartes/de formulario bien formados.	

Código de estado HTTP	Código de error	Descripción del código de error	
	MalformedXML	El XML proporcionado no tenía un formato válido o no podía validarse con nuestro esquema publicado.	
	MaxPostPreDataLengthExceededError	Los campos de solicitud POST que preceden al archivo de carga eran demasiado grandes.	
	MetadataTooLarge	Los encabezados de los metadatos superan el tamaño máximo de metadatos permitido.	
	MissingRequestBodyError	Ha enviado un documento XML vacío como una solicitud.	
	MissingSecurityHeader	Falta un encabezado obligatorio en la solicitud.	
	NoLoggingStatusForKey	No existe un subrecurso de estado de registro para una clave.	
	RequestHeaderSectionTooLarge	El encabezado de la solicitud y los parámetros de consulta utilizados para realizar la solicitud superan los tamaños máximos permitidos	

Código de estado HTTP	Código de error	Descripción del código de error
	UnexpectedContent	Esta solicitud tiene contenido no compatible.
	UserKeyMustBeSpecified	La solicitud POST del bucket debe contener el nombre de campo especificado. Si se especifica, compruebe el orden de los campos.
	IncorrectEndpoint	El bucket especificado existe en otra región. Dirija las solicitudes al punto de conexión correcto.
403: prohibido	RequestTimeTooSkewed	La diferencia entre la hora de la solicitud y la hora del servidor es demasiado grande.
	SignatureDoesNotMatch	La firma de solicitud que calculó el servidor no coincide con la firma que proporcionó. Compruebe la clave de acceso secreta de AWS y el método de firma. Para obtener más información, consulte Autenticación de REST y Autenticación de SOAP.

Código de estado HTTP	Código de error	Descripción del código de error	
	NotSignedUp	La cuenta no está inscrita para el servicio de Amazon S3. Debe registrarse antes de poder utilizar Amazon S3. Puede registrarse en la siguiente URL: https://aws.amazon.com/s3	
	InvalidSecurity	Las credenciales de seguridad proporcionadas no son válidas.	
	InvalidPayer	Se ha desactivado todo el acceso a este objeto. Para obtener más ayuda, consulte Contacte con nosotros.	
	InvalidAccessKeyId	El ID de clave de acceso de AWS proporcionado no existe en nuestros registros.	
	AccountProblem	Hay un problema con la Cuenta de AWS que evita que la operación se complete de forma satisfactoria. Para obtener más ayuda, consulte Contacte con nosotros.	

Código de estado HTTP	Código de error	Descripción del código de error	
	UnauthorizedAccessError	Aplicable solo en las regiones de China. Se devuelve cuando se hace una solicitud a un bucket que no tiene una licencia ICP. Para obtener más información, consulte Registros de ICP.	
404 Not Found (No encontrado)	NoSuchUpload	La carga multiparte especificada no existe. Es posible que el ID de carga no sea válido o que la carga multiparte se haya cancelado o completado.	
	NoSuchWebsiteConfiguration	El bucket especificado no tiene una configuración de sitio web.	
Método no permitido, 405	MethodNotAllowed	El método especificado no está permitido en este recurso.	

Código de estado HTTP	Código de error	Descripción del código de error	
Conflicto, 409	BucketAlreadyExists	El nombre del bucket solicitado no está disponible. Todos los usuarios del sistema comparten el espacio de nombres del bucket. Especifique un nombre diferente e inténtelo de nuevo.	
	InvalidBucketState	La solicitud no es válida para el estado actual del bucket.	
	OperationAborted	Una operación condicional en conflicto está en curso actualmente en este recurso. Inténtelo de nuevo.	
Longitud requerida, 411	MissingContentLength	Debe proporcionar el encabezado HTTP de longitud del contenido.	
Condición previa con error, 412	RequestIsNotMultiPartContent	Una solicitud POST de bucket debe ser del tipo de documento adjunto multiparte o datos de formulario.	

Filtrado y recuperación de datos con Amazon S3 Select

> **Important**
>
> Amazon S3 Select ya no está disponible para los nuevos clientes. Los clientes actuales de Amazon S3 Select pueden seguir utilizando la característica de la forma habitual. Más información

Amazon S3 Select le permite utilizar instrucciones de lenguaje de consulta estructurada (SQL) para filtrar el contenido de los objetos de Amazon S3 y recuperar exactamente el subconjunto de datos que necesita. Si utiliza Amazon S3 Select para filtrar estos datos, puede reducir la cantidad de datos que Amazon S3 transfiere, lo que reduce también los costos y la latencia para recuperarlos.

Amazon S3 Select solo permite consultar un objeto a la vez. Funciona con un objeto almacenado en formato CSV, JSON o Apache Parquet. También funciona con objetos comprimidos con GZIP o BZIP2 (solo para objetos CSV y JSON), así como con un objeto cifrado del lado del servidor. Puede especificar el formato de los resultados como CSV o JSON, y también puede determinar cómo se delimitan los registros en los resultados.

Las expresiones SQL se pasan a Amazon S3 en la solicitud. Amazon S3 Select es compatible con un subconjunto de SQL. Para obtener más información sobre los elementos SQL compatibles con Amazon S3 Select, consulte Referencia de SQL para Amazon S3 Select.

Puede realizar consultas de SQL con la consola de Amazon S3, la AWS Command Line Interface (AWS CLI), la operación de la API de REST `SelectObjectContent` o los SDK de AWS.

> **Note**
>
> La consola de Amazon S3 limita la cantidad de datos devueltos a 40 MB. Para recuperar más datos, utilice la AWS CLI o la API.

Requisitos y límites

A continuación, se describen los requisitos para utilizar Amazon S3 Select:

- Debe tener el permiso `s3:GetObject` para el objeto que está consultando.

- Si el objeto que está consultando está cifrado con cifrado en el servidor con claves proporcionados por el cliente (SSE-C), debe utilizar `https` y proporcionar la clave de cifrado en la solicitud.

Cuando se usa Amazon S3 Select, se aplican los siguientes límites:

- S3 Select solo puede consultar un objeto por solicitud.
- La longitud máxima de las expresiones SQL es de 256 KB.
- La longitud máxima de un registro en la entrada o resultado es de 1 MB.
- Amazon S3 Select solo puede emitir datos anidados mediante el formato de salida de JSON.
- No se pueden consultar un objeto almacenado en las clases de almacenamiento S3 Glacier Flexible Retrieval, S3 Glacier Deep Archive o Reduced Redundancy Storage (RRS). Tampoco puede consultar un objeto almacenado en el nivel S3 Intelligent-Tiering Archive Access ni en el nivel S3 Intelligent-Tiering Deep Archive Access. Para obtener más información acerca de las clases de almacenamiento, consulte Uso de las clases de almacenamiento de Amazon S3.

Se aplican limitaciones adicionales al utilizar Amazon S3 Select con un objeto Parquet:

- Amazon S3 Select solo admite compresión en columnas usando GZIP o Snappy. Amazon S3 Select no admite la compresión de todo el objeto para un objeto Parquet.
- Amazon S3 Select no admite salida Parquet. Debe especificar el formato de salida como CSV o JSON.
- El tamaño de grupo de filas sin comprimir máximo es de 512 MB.
- Debe utilizar los tipos de datos que están especificados en el esquema del objeto.
- Seleccionar en un campo repetido devuelve solo el último valor.

Crear una solicitud

Cuando se construye una solicitud, se deben proporcionar detalles del objeto que se está consultando mediante un objeto `InputSerialization`. También utilizará un objeto `0utputSerialization` para proporcionar información sobre cómo se deben devolver los resultados. Asimismo, deberá incluir la expresión SQL que Amazon S3 utiliza para filtrar la solicitud.

Para obtener más información acerca de cómo construir una solicitud de selección de Amazon S3, consulte SelectObjectContent en la referencia de API de Amazon Simple Storage Service. También puede consultar uno de los ejemplos de código de SDK en las secciones siguientes.

Solicitudes que utilizan intervalos de análisis

Amazon S3 Select le permite analizar un subconjunto de un objeto especificando el intervalo de bytes que se desea consultar. Esto le permite paralelizar el análisis del objeto completo dividiendo el trabajo en solicitudes separadas de Amazon S3 Select para una serie de intervalos de análisis sin solapamiento.

No es necesario que los intervalos de análisis estén alineados con los límites de registros. Una solicitud de intervalo de análisis de Amazon S3 Select se ejecuta en el intervalo de bytes especificado. La consulta procesará un registro que comienza dentro del intervalo de análisis especificado, pero se extiende más allá de ese intervalo de análisis. Por ejemplo, a continuación se muestra un objeto de Amazon S3 que contiene una serie de registros en formato CSV delimitado por líneas:

```
A,B
C,D
D,E
E,F
G,H
I,J
```

Suponga que utiliza el parámetro `ScanRange` de Amazon S3 Select y establece Inicio en (Byte) 1 y Fin en (Byte) 4. De este modo, el rango de escaneo comenzaría en "`,`" y analizaría hasta el final del registro a partir de`C`. Su solicitud de rango de escaneo devolverá el resultado `C, D` porque ese es el final del registro.

El rango de análisis de Amazon S3 Select es compatible con objetos Parquet, CSV (sin delimitadores entrecomillados) o JSON (solo en modo `LINES`). Los objetos CSV y JSON deben estar sin comprimir. Para objetos JSON y CSV basados en líneas, cuando se especifica un intervalo de análisis como parte de la solicitud de Amazon S3 Select, se procesan todos los registros que comienzan dentro del intervalo de análisis. Para objetos Parquet, se procesan todos los grupos de filas que comienzan dentro del rango de análisis solicitado.

Las solicitudes de intervalo de exploración de Amazon S3 Select están disponibles para utilizarlas con la AWS CLI, la API de Amazon S3 y los SDK de AWS. Puede utilizar el parámetro `ScanRange` en la solicitud de Amazon S3 Select para esta funcionalidad. Para obtener más información, consulte SelectObjectContent en la Referencia de la API de Amazon Simple Storage Service.

Errores

Amazon S3 Select devuelve un código de error y un mensaje de error asociado cuando se produce un problema al intentar ejecutar una consulta. Para obtener una lista de códigos de error y descripciones, consulte la sección Lista de códigos de error de contenido de objetos SELECT de la página Respuestas de error en la referencia de Amazon Simple Storage Service API.

Para obtener más información acerca de Amazon S3 Select, consulte los siguientes temas.

Temas

- Ejemplos de uso de Amazon S3 Select en un objeto
- Referencia de SQL para Amazon S3 Select

Ejemplos de uso de Amazon S3 Select en un objeto

> **Important**
>
> Amazon S3 Select ya no está disponible para los nuevos clientes. Los clientes actuales de Amazon S3 Select pueden seguir utilizando la característica de la forma habitual. Más información

Puede utilizar S3 Select para seleccionar contenido de un objeto con la consola de Amazon S3, la API de REST y los SDK de AWS.

Para obtener más información sobre las funciones de SQL admitidas para S3 Select, consulte Funciones SQL.

Uso de la consola de S3

Para seleccionar contenido de un objeto en la consola de Amazon S3

1. Inicie sesión AWS Management Console Management Console y abra la consola de Amazon S3 en https://console.aws.amazon.com/s3/.
2. En el panel de navegación izquierdo, elija Instancias.
3. Elija el bucket que contiene el objeto del que desea seleccionar el contenido y, a continuación, elija el nombre del objeto.
4. Elija Acciones de objetos y Consultar con S3 Select.

5. Configure Configuración de entrada en función del formato de los datos de entrada.
6. Configure Configuración de salida en función del formato de la salida que desee recibir.
7. Para extraer registros del objeto elegido, en Consulta SQL, introduzca los comandos SELECT SQL. Para obtener más información sobre cómo escribir comandos SQL, consulte Referencia de SQL para Amazon S3 Select.
8. Después de introducir las consultas SQL, seleccione Ejecutar consulta SQL. A continuación, en Resultados de la consulta, puede ver los resultados de sus consultas SQL.

Uso de la API de REST

Puede utilizar los SDK de AWS para seleccionar contenido de un objeto. Sin embargo, si su aplicación lo requiere, puede enviar solicitudes REST directamente. Para obtener más información sobre el formato de solicitud y respuesta, consulte SelectObjectContent.

Uso de los AWS SDK

Puede utilizar Amazon S3 Select para seleccionar parte del contenido de un objeto mediante el método `selectObjectContent`. Si este método funciona correctamente, devuelve los resultados de la expresión SQL.

Java

En el siguiente ejemplo de código Java se devuelve el valor de la primera columna de cada uno de los registros almacenados en un objeto que contiene datos en formato CSV. También se solicita la devolución de mensajes `Progress` y `Stats`. Se debe proporcionar un nombre de bucket válido y un objeto que contenga datos en formato CSV.

Para obtener instrucciones sobre cómo crear y probar una muestra funcional, consulte Introducción en la Guía del desarrollador de AWS SDK for Java.

```
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

Para obtener un ejemplo de JavaScript que utiliza AWS SDK for JavaScript con la operación de la API de S3 `SelectObjectContent` para seleccionar registros de archivos JSON y CSV almacenados en Amazon S3, consulte la publicación de blog Introducción de ayuda para Amazon S3 Select en AWS SDK for JavaScript.

Python

Para obtener un ejemplo de Python sobre el uso de consultas SQL para buscar en los datos que se han cargado en Amazon S3 como un archivo de valores separados por comas (CSV) mediante S3 Select, consulte la publicación de blog Consulta de datos sin servidores o bases de datos mediante Amazon S3 Select.

Referencia de SQL para Amazon S3 Select

> **Important**
>
> Amazon S3 Select ya no está disponible para los nuevos clientes. Los clientes actuales de Amazon S3 Select pueden seguir utilizando la característica de la forma habitual. Más información

Esta referencia contiene una descripción de los elementos del lenguaje de consulta estructurada (SQL, Structured Query Language) compatibles con Amazon S3 Select.

Temas

- SELECT command

- Tipos de datos
- Operadores
- Palabras clave reservadas
- Funciones SQL

SELECT command

> **Important**
>
> Amazon S3 Select ya no está disponible para los nuevos clientes. Los clientes actuales de Amazon S3 Select pueden seguir utilizando la característica de la forma habitual. Más información

El trabajo Select de solamente es compatible con el comando `SELECT` de SQL. Las siguientes cláusulas del estándar ANSI son compatibles con `SELECT`:

- `SELECT` list
- `FROM`Cláusula
- Cláusula `WHERE`
- `LIMIT`Cláusula

> **Note**
>
> Actualmente, las consultas de Amazon S3 Select no admiten subconsultas ni combinaciones.

SELECT list

La lista `SELECT` asigna un nombre a las columnas, funciones y expresiones que desea que devuelva la consulta. La lista representa el resultado de la consulta.

```
SELECT *
SELECT projection1 AS column_alias_1, projection2 AS column_alias_2
```

El primer formato de SELECT con * (asterisco) devuelve todas las filas que superan la cláusula WHERE, tal y como están. La segunda forma de SELECT crea una fila con expresiones escalares de salida definidas por el usuario ***projection1*** y ***projection2*** para cada columna.

Cláusula FROM

Amazon S3 Select admite los siguientes formatos en la cláusula FROM:

```
FROM table_name
FROM table_name alias
FROM table_name AS alias
```

En cada forma de la cláusula FROM, table_name es el S3Object que se está consultando. Los usuarios acostumbrados a las bases de datos relacionales tradicionales pueden hacerse a la idea de que se trata de un esquema de base de datos que contiene varias vistas de una tabla.

Siguiendo el código SQL estándar, la cláusula FROM crea filas que se filtran en la cláusula WHERE y se proyectan en la lista SELECT.

Para objetos JSON almacenados en Amazon S3 Select, también puede usar las siguientes formas de la cláusula FROM:

```
FROM S3Object[*].path
FROM S3Object[*].path alias
FROM S3Object[*].path AS alias
```

Con esta forma de la cláusula FROM, puede seleccionar de matrices u objetos dentro de un objeto JSON. Puede especificar path, mediante una de las formas siguientes:

- Por nombre (en un objeto): .*name* o ['*name*']
- Por índice (en una matriz): [*index*]
- Por carácter comodín (en un objeto): .*
- Por carácter comodín (en una matriz): [*]

> **Note**
>
> - Esta forma de la cláusula FROM solo funciona con objetos JSON.

- Los caracteres comodín siempre emiten al menos un registro. Si no coincide ningún registro, Amazon S3 Select emite el valor `MISSING`. Durante la serialización de salida (después de que finalice la ejecución de la consulta), Amazon S3 Select reemplaza los valores de `MISSING` por registros vacíos.
- Las funciones de agregación (`AVG`, `COUNT`, `MAX`, `MIN` y `SUM`) hacen caso omiso de los valores de `MISSING`.
- Si no proporciona un alias cuando usa un carácter comodín, puede referirse a la fila mediante el último elemento en la ruta. Por ejemplo, puede seleccionar todos los precios de una lista de libros mediante la consulta `SELECT price FROM S3Object[*].books[*].price`. Si la ruta termina con un carácter comodín en lugar de un nombre, puede usar el valor `_1` para referirse a la fila. Por ejemplo, en lugar de `SELECT price FROM S3Object[*].books[*].price`, podría utilizar la consulta `SELECT _1.price FROM S3Object[*].books[*]`.
- Amazon S3 Select siempre trata un documento JSON como una matriz de valores de nivel raíz. Por lo tanto, incluso si el objeto JSON que está consultando solo tiene un elemento raíz, la cláusula `FROM` debe comenzar con `S3Object[*]`. Sin embargo, por razones de compatibilidad, Amazon S3 Select le permite omitir el carácter comodín si no incluye una ruta. Además, la cláusula completa `FROM S3Object` es equivalente a `FROM S3Object[*] as S3Object`. Si incluye una ruta, también debe usar el carácter comodín. Por lo tanto, `FROM S3Object` y `FROM S3Object[*].`*`path`* son cláusulas válidas, pero `FROM S3Object.`*`path`* no.

Example

Ejemplos:

Ejemplo 1

Este ejemplo muestra resultados al usar el conjunto de datos y la consulta siguientes:

```
{ "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 produce cada resultado por las siguientes razones:

- `{"id":"id-1"}`: `S3Object[0].Rules[0].id` produjo una coincidencia.
- `{}`: `S3Object[0].Rules[1].id` no encontró un registro coincidente, por lo que Amazon S3 Select emitió `MISSING`, que se cambió a un registro vacío durante la serialización de salida y se devolvió.
- `{"id":"id-2"}`: `S3Object[0].Rules[2].id` produjo una coincidencia.
- `{}`: `S3Object[1]` no encontró un registro coincidente en `Rules`, por lo que Amazon S3 Select emitió `MISSING`, que se cambió a un registro vacío durante la serialización de salida y se devolvió.

Si no quiere que Amazon S3 Select devuelva registros vacíos cuando no encuentre una coincidencia, puede probar el valor `MISSING`. La siguiente consulta devuelve los mismos resultados que la consulta anterior, pero con los valores vacíos omitidos:

```
SELECT id FROM S3Object[*].Rules[*].id WHERE id IS NOT MISSING
```

```
{"id":"1"}
{"id":"2"}
```

Ejemplo 2

Este ejemplo muestra resultados al usar el conjunto de datos y las consultas siguientes:

```
{ "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"}
```

Cláusula WHERE

La cláusula `WHERE` utiliza esta sintaxis:

```
WHERE condition
```

La cláusula `WHERE` filtra las filas en función de *`condition`*. Una condición es una expresión que tiene un resultado booleano. En el resultado, solamente se devuelven las filas en las que la condición es `TRUE`.

Cláusula LIMIT

La cláusula `LIMIT` utiliza esta sintaxis:

```
LIMIT number
```

La cláusula `LIMIT` limita el número de registros que debe devolver la consulta basándose en *`number`*.

Acceso mediante atributos

Las cláusulas `SELECT` y `WHERE` pueden referirse a los datos de los registros mediante uno de los métodos de las secciones siguientes, dependiendo de si el archivo que se está consultando está en formato CSV o en formato JSON.

CSV

- Números de columnas: se puede hacer referencia a la columna n de una fila con el nombre de columna _*N*, donde *N* es la posición de la columna. El número de posición empieza en 1. Por ejemplo, la primera columna se denomina _1 y la segunda, _2.

Se puede hacer referencia a una columna como *_N* o *alias._N*. Por ejemplo, _2 y myAlias._2 son formas válidas de hacer referencia a una columna en la lista SELECT y la cláusula WHERE.

- Encabezados de columna: para los objetos con formato CSV que tienen una fila de encabezado, los encabezados están disponibles para la lista SELECT y la cláusula WHERE. En concreto, al igual que ocurre en SQL tradicional, dentro de las expresiones de las cláusulas SELECT y WHERE, se puede hacer referencia a las columnas mediante *alias.column_name* o *column_name*.

JSON

- Documento: se puede tener acceso a los campos del documento JSON como *alias.name*. También puede acceder a los campos anidados, por ejemplo, *alias.name1.name2.name3*.
- Lista: se puede obtener acceso a los elementos de una lista JSON mediante índices basados en cero con el operador []. Por ejemplo, se puede obtener acceso al segundo elemento de una lista como *alias*[1]. Puede combinar el acceso a los elementos de la lista con campos, por ejemplo,*alias.name1.name2*[1].*name3*.
- Ejemplos: considere este objeto JSON como un conjunto de datos de ejemplo:

```
{"name": "Susan Smith",
"org": "engineering",
"projects":
    [
     {"project_name":"project1", "completed":false},
     {"project_name":"project2", "completed":true}
    ]
}
```

Ejemplo 1

La siguiente consulta devuelve estos resultados:

```
Select s.name from S3Object s
```

```
{"name":"Susan Smith"}
```

Ejemplo 2

La siguiente consulta devuelve estos resultados:

```
Select s.projects[0].project_name from S3Object s
```

```
{"project_name":"project1"}
```

Distinción entre mayúsculas y minúsculas en los nombres de atributos o encabezados

Con Amazon S3 Select, puede utilizar comillas dobles para indicar que los encabezados de columna (para los objetos CSV) y los atributos (para los objetos JSON) distinguen entre mayúsculas y minúsculas. Sin comillas dobles, los encabezados y los atributos de los objetos no distinguen entre mayúsculas y minúsculas. En casos de ambigüedad, se produce un error.

Los ejemplos siguientes son: 1) objetos de Amazon S3 Select en formato CSV con los encabezados de columna especificados y con `FileHeaderInfo` establecido a `"Use"` para la solicitud de consulta; o 2) objetos de Amazon S3 en formato JSON con los atributos especificados.

Ejemplo 1: el objeto que se consulta tiene el encabezado o el atributo `NAME`.

- La expresión siguiente devuelve los valores del objeto correctamente. Como no hay comillas, la consulta no distingue entre mayúsculas y minúsculas.

```
SELECT s.name from S3Object s
```

- La expresión siguiente da como resultado un error 400 `MissingHeaderName`. Como hay comillas, la consulta distingue entre mayúsculas y minúsculas.

```
SELECT s."name" from S3Object s
```

Ejemplo 2: el objeto de Amazon S3 que se consulta tiene un encabezado o un atributo con `NAME` y otro encabezado o atributo con `name`.

- La expresión siguiente da como resultado un error 400 `AmbiguousFieldName`. Como no hay comillas, la consulta no distingue entre mayúsculas y minúsculas, pero hay dos coincidencias, por lo que se genera el error.

```
SELECT s.name from S3Object s
```

- La expresión siguiente devuelve los valores del objeto correctamente. Como hay comillas, la consulta distingue entre mayúsculas y minúsculas, por lo que no hay ambigüedad.

```
SELECT s."NAME" from S3Object s
```

Uso de palabras clave reservadas como términos definidos por el usuario

Amazon S3 Select tiene un conjunto de palabras clave reservadas que son necesarias para ejecutar las expresiones SQL utilizadas para consultar el contenido de los objetos. Entre estas palabras clave reservadas se incluyen nombres de funciones, tipos de datos, operadores, etc. En algunos casos, los términos definidos por el usuario, como los encabezados de columna (para los archivos CSV) o los atributos (para los objetos JSON), pueden entrar en conflicto con una palabra clave reservada. Cuando esto ocurre, debe utilizar comillas dobles para indicar que está utilizando deliberadamente un término definido por el usuario que entra en conflicto con una palabra clave reservada. De lo contrario, se producirá un error de análisis 400.

Para obtener la lista completa de las palabras clave reservadas, consulte Palabras clave reservadas.

El ejemplo siguiente es: 1) un objeto de Amazon S3 Select en formato CSV con los encabezados de columna especificados y con `FileHeaderInfo` establecido a `"Use"` para la solicitud de consulta; o 2) un objeto de Amazon S3 en formato JSON con los atributos especificados.

Ejemplo: el objeto que se consulta tiene el encabezado o el atributo denominado `CAST`, que es una palabra clave reservada.

- La expresión siguiente devuelve los valores del objeto correctamente. Como se utilizan comillas en la consulta, S3 Select utiliza el encabezado o el atributo definido por el usuario.

```
SELECT s."CAST" from S3Object s
```

- La expresión siguiente da como resultado un error de análisis 400. Como no se utilizan comillas en la consulta, `CAST` está en conflicto con una palabra clave reservada.

```
SELECT s.CAST from S3Object s
```

Expresiones escalares

En la cláusula `WHERE` y la lista `SELECT`, puede tener expresiones escalares de SQL, que son expresiones que devuelven valores escalares. Tienen el siguiente formato:

- ***literal***

  Literal SQL.
- ***column_reference***

  Una referencia a una columna con el formato *column_name* o *alias*`.`*column_name*.
- ***unary_op expression***

  En este caso, ***unary_op*** es un operador unario de SQL.
- ***expression binary_op expression***

  En este caso, ***binary_op*** es un operador binario de SQL.
- ***func_name***

  En este caso, ***func_name*** es el nombre de la función escalar que se va a invocar.
- ***expression*** `[ NOT ] BETWEEN` ***expression*** `AND` ***expression***
- ***expression*** `LIKE` ***expression*** `[ ESCAPE` ***expression*** `]`

Tipos de datos

> **Important**
>
> Amazon S3 Select ya no está disponible para los nuevos clientes. Los clientes actuales de Amazon S3 Select pueden seguir utilizando la característica de la forma habitual. Más información

Amazon S3 Select y S3 Glacier Select admite varios tipos de datos primitivos.

Conversiones de tipos de datos

Por lo general, suele seguirse la función `CAST`, si está definida. Si no se ha definido `CAST`, todos los datos de entrada se tratarán como una cadena. En ese caso, deberá convertir sus datos de entrada en los tipos de datos pertinentes cuando sea necesario.

Para obtener más información sobre la función CAST, consulte CAST.

Tipos de datos compatibles

Amazon S3 Select admite el siguiente subconjunto de tipos de datos primitivos.

Nombre	Descripción	Ejemplos
bool	Un valor booleano, ya sea TRUE o FALSE.	FALSE
int, integer	Un entero con signo de 8 bytes comprendido entre -9 223 372 036 854 775 808 y 9 223 372 036 854 775 807.	100000
string	Una cadena de longitud variable codificada en UTF8. El límite predeterminado es de 1 carácter. El límite máximo de caracteres es de 2 147 483 647.	'xyz'
float	Un número de punto flotante de 8 bits.	CAST(0.456 AS FLOAT)
decimal, numeric	Un número en base 10, con una precisión máxima de 38 (es decir, el número máximo de dígitos significativos), y con una escala en un intervalo de entre -2^{31} y $2^{31}-1$ (es decir, el exponente en base 10). **Note** Amazon S3 Select ignora la escala y la precisión cuando se proporcionan ambos al mismo tiempo.	123.456
timestamp	Las marcas temporales representan un momento concreto, siempre incluyen el desfase horario local y permiten establecer una precisión arbitraria. En formato de texto, las marcas temporales siguen los formatos de fecha y hora de la notación W3C, pero deben terminar por el literal T si las marcas temporales no tienen, como mínimo, una precisión de un día completo. Se pueden	CAST('2007-04-05T14:30Z' AS TIMESTAMP)

Nombre	Descripción	Ejemplos
	utilizar fracciones de segundo con al menos un dígito de precisión y sin ningún límite máximo. El desfase horario local puede representarse con el formato hora:minuto con relación a UTC o con el literal `Z` para indicar la hora local en UTC. Los desfases horarios locales deben incluirse en las marcas temporales que contienen la hora, pero no están permitidos en los valores de fecha.	

Tipos de Parquet admitidos

Amazon S3 Select admite los siguientes tipos de Parquet.

- `DATE`
- `DECIMAL`
- `ENUM`
- `INT(8)`
- `INT(16)`
- `INT(32)`
- `INT(64)`
- `LIST`

> **Note**
>
> Para la salida de tipo Parquet `LIST`, Amazon S3 Select solo admite el formato JSON. Sin embargo, si la consulta limita los datos a valores simples, también se puede consultar el tipo Parquet `LIST` en formato CSV.

- `STRING`
- Precisión admitida de `TIMESTAMP` (`MILLIS`/`MICROS`/`NANOS`)

> **Note**
>
> No se admiten las marcas temporales guardadas como `INT(96)`.

Debido al rango del tipo `INT(64)`, las marcas temporales que utilizan la unidad `NANOS` solo pueden representar valores entre `1677-09-21 00:12:43` y `2262-04-11 23:47:16`. Los valores fuera de este rango no se pueden representar con la unidad `NANOS`.

Asignación de tipos de Parquet a tipos de datos admitidos en Amazon S3 Select

Tipos de Parquet	Tipos de datos compatibles
`DATE`	`timestamp`
`DECIMAL`	`decimal`, `numeric`
`ENUM`	`string`
`INT(8)`	`int`, `integer`
`INT(16)`	`int`, `integer`
`INT(32)`	`int`, `integer`
`INT(64)`	`decimal`, `numeric`
`LIST`	Cada tipo de Parquet de la lista se asigna al tipo de datos correspondiente.
`STRING`	`string`
`TIMESTAMP`	`timestamp`

Operadores

> **Important**
>
> Amazon S3 Select ya no está disponible para los nuevos clientes. Los clientes actuales de Amazon S3 Select pueden seguir utilizando la característica de la forma habitual. Más información

Amazon S3 Select admite los siguientes operadores.

Logical operators (Operadores lógicos)

- `AND`
- `NOT`
- `OR`

Operadores de comparación

- `<`
- `>`
- `<=`
- `>=`
- `=`
- `<>`
- `!=`
- `BETWEEN`
- `IN` – Por ejemplo: `IN ('a', 'b', 'c')`

Operadores de coincidencia de patrones

- `LIKE`
- `_` (coincide con cualquier carácter)

- `%` (coincide con cualquier secuencia de caracteres)

Operadores unitarios

- `IS NULL`
- `IS NOT NULL`

Operadores matemáticos

Se admiten los operadores de suma, resta, multiplicación, división y módulo:

- `+`
- `-`
- `*`
- `/`
- `%`

Jerarquía de los operadores

En la siguiente tabla se muestra la prioridad de los operadores en orden descendente.

Operador o elemento	Asociatividad	Obligatorio
`-`	derecha	menos unario
`*`, `/`, `%`	izquierda	multiplicación, división, módulo
`+`, `-`	izquierda	suma, resta
`IN`		pertenencia a un conjunto
`BETWEEN`		limitación de intervalos

Operador o elemento	Asociatividad	Obligatorio
`LIKE`		coincidencia de patrones de cadena
`<>`		menor que, mayor que
`=`	derecha	igualdad, asignación
`NOT`	derecha	negación lógica
`AND`	izquierda	conjunción lógica
`OR`	izquierda	disyunción lógica

Palabras clave reservadas

> **Important**
>
> Amazon S3 Select ya no está disponible para los nuevos clientes. Los clientes actuales de Amazon S3 Select pueden seguir utilizando la característica de la forma habitual. Más información

A continuación, se muestra la lista de palabras clave reservadas para Amazon S3 Select. Entre estas palabras clave se incluyen los nombres de funciones, tipos de datos, operadores, etc., que se necesitan para ejecutar las expresiones SQL utilizadas para consultar el contenido de los objetos.

```
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
```

Funciones SQL

> **Important**
>
> Amazon S3 Select ya no está disponible para los nuevos clientes. Los clientes actuales de Amazon S3 Select pueden seguir utilizando la característica de la forma habitual. Más información

Amazon S3 Select admite las siguientes funciones SQL.

Temas

- Funciones de agregación
- Funciones condicionales
- Funciones de conversión
- Funciones de datos
- Funciones de cadena

Funciones de agregación

> **Important**
>
> Amazon S3 Select ya no está disponible para los nuevos clientes. Los clientes actuales de Amazon S3 Select pueden seguir utilizando la característica de la forma habitual. Más información

Amazon S3 Select admite las siguientes funciones de agregación.

Función	Tipo de argumento	Tipo de retorno
`AVG(`*expressio*`n )`	`INT`, `FLOAT`, `DECIMAL`	`DECIMAL` para un argumento `INT`, `FLOAT` un argumento de coma flotante; en el resto de casos, el mismo que el tipo de datos del argumento.
`COUNT`	-	`INT`
`MAX(`*expressio*`n )`	`INT`, `DECIMAL`	El mismo que el tipo del argumento.
`MIN(`*expressio*`n )`	`INT`, `DECIMAL`	El mismo que el tipo del argumento.
`SUM(`*expressio*`n )`	`INT`, `FLOAT`, `DOUBLE`, `DECIMAL`	`INT` para un argumento `INT`, `FLOAT` un argumento de coma

Función	Tipo de argumento	Tipo de retorno
		flotante; en el resto de casos, el mismo que el tipo de datos del argumento.

SUMEjemplo de

Para agregar los tamaños totales de los objetos de una carpeta en un informe de S3 Inventary, utilice una expresión `SUM`.

El siguiente informe de inventario de S3 es un archivo CSV comprimido con GZIP. Hay tres columnas.

- La primera columna es el nombre del bucket de S3 (*`DOC-EXAMPLE-BUCKET`*) al que se destina el informe de inventario de S3.
- La segunda columna es el nombre clave que identifica de forma exclusiva el objeto en el bucket.

  El valor *`example-folder/`* de la primera fila corresponde a la carpeta *`example-folder`*. En Amazon S3, cuando crea una carpeta en su bucket, S3 crea un objeto de 0 bytes con una clave establecida al nombre de la carpeta que ha proporcionado.

  El valor *`example-folder/object1`* de la segunda fila corresponde al objeto *`object1`* de la carpeta *`example-folder`*.

  El valor *`example-folder/object2`* de la tercera fila corresponde al objeto *`object2`* de la carpeta *`example-folder`*.

  Para obtener más información sobre las carpetas de S3, consulte Organización de objetos en la consola de Amazon S3 con carpetas.
- La tercera columna es el tamaño del objeto en bytes.

```
"DOC-EXAMPLE-BUCKET","example-folder/","0"
"DOC-EXAMPLE-BUCKET","example-folder/object1","2011267"
"DOC-EXAMPLE-BUCKET","example-folder/object2","1570024"
```

Para usar una expresión SUM para calcular el tamaño total de la carpeta *example-folder*, ejecute la consulta SQL con Amazon S3 Select.

```
SELECT SUM(CAST(_3 as INT)) FROM s3object s WHERE _2 LIKE 'example-folder/%' AND _2 !=
 'example-folder/';
```

Resultado de la consulta:

```
3581291
```

Funciones condicionales

> **Important**
>
> Amazon S3 Select ya no está disponible para los nuevos clientes. Los clientes actuales de Amazon S3 Select pueden seguir utilizando la característica de la forma habitual. Más información

Amazon S3 Select admite las siguientes funciones condicionales.

Temas

- CASE
- COALESCE
- NULLIF

CASE

La expresión CASE es una expresión condicional similar a las instrucciones `if/then/else` que se encuentran en otros lenguajes. CASE se utiliza para especificar un resultado cuando hay varias condiciones. Existen dos tipos de expresiones CASE: simple y buscada.

En expresiones CASE simples, una expresión se compara con un valor. Cuando hay una coincidencia, se aplica la acción especificada en la cláusula THEN. Si no se encuentra ninguna coincidencia, se aplica la acción en la cláusula ELSE.

En las expresiones CASE buscadas, cada CASE se evalúa según una expresión booleana y la instrucción CASE devuelve el primer CASE que coincida. Si no hay ninguna coincidencia de CASE entre las cláusulas WHEN, se devuelve la acción en la cláusula ELSE.

Sintaxis

> **Note**
>
> Actualmente, Amazon S3 Select no admite ORDER BY ni consultas que contienen líneas nuevas. Asegúrese de usar consultas sin saltos de línea.

A continuación, se muestra una instrucción CASE sencilla que se utiliza para hacer coincidir condiciones:

```
CASE expression WHEN value THEN result [WHEN...] [ELSE result] END
```

La siguiente es una instrucción CASE buscada que se utiliza para evaluar cada condición:

```
CASE WHEN boolean condition THEN result [WHEN ...] [ELSE result] END
```

Ejemplos

> **Note**
>
> Si utiliza la consola de Amazon S3 para ejecutar los siguientes ejemplos y el archivo CSV contiene una fila de encabezado, elija Excluir la primera línea de datos CSV.

Ejemplo 1: utilice una expresión CASE simple para reemplazar New York City por Big Apple en una consulta. Reemplace todos los demás nombres de ciudad por other.

```
SELECT venuecity, CASE venuecity WHEN 'New York City' THEN 'Big Apple' ELSE 'other' END
 FROM S3Object;
```

Resultado de la consulta:

```
venuecity       |   case
----------------+-----------
```

```
Los Angeles      | other
New York City    | Big Apple
San Francisco    | other
Baltimore        | other
...
```

Ejemplo 2: Utilice una expresión CASE buscada para asignar números de grupo según el valor `pricepaid` para ventas de tickets individuales:

```
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;
```

Resultado de la consulta:

```
pricepaid |  case
----------+---------
12624.00 | group 2
10000.00 | group 3
10000.00 | group 3
9996.00 | group 1
9988.00 | group 1
...
```

COALESCE

`COALESCE` evalúa los argumentos por orden y devuelve el primero que no sea desconocido, es decir, el primero no nulo o que no falta. Esta función no propaga los argumentos nulos o que faltan.

Sintaxis

```
COALESCE ( expression, expression, ... )
```

Parámetros

`expression`

La expresión de destino sobre la que opera la función.

Ejemplos

```
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

Dadas dos expresiones, `NULLIF` devuelve `NULL` si ambas toman el mismo valor; en caso contrario, `NULLIF` devuelve el resultado de la evaluación de la primera expresión.

Sintaxis

```
NULLIF ( expression1, expression2 )
```

Parámetros

expression1, *expression2*

Las expresiones de destino sobre las que opera la función.

Ejemplos

```
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
```

Funciones de conversión

> **Important**
>
> Amazon S3 Select ya no está disponible para los nuevos clientes. Los clientes actuales de Amazon S3 Select pueden seguir utilizando la característica de la forma habitual. Más información

Amazon S3 Select admite la siguiente función de conversión.

Temas

- CAST

CAST

La función `CAST` convierte una entidad (por ejemplo, una expresión que da como resultado un único valor) de un tipo a otro.

Sintaxis

```
CAST ( expression AS data_type )
```

Parámetros

`expression`

Combinación de uno o varios valores, operadores o funciones SQL que dan como resultado un valor.

`data_type`

Tipo de datos de destino (por ejemplo, `INT`) al que se va a convertir la expresión. Para obtener una lista de los tipos de datos admitidos, consulte Tipos de datos.

Ejemplos

```
CAST('2007-04-05T14:30Z' AS TIMESTAMP)
```

```
CAST(0.456 AS FLOAT)
```

Funciones de datos

> **Important**
>
> Amazon S3 Select ya no está disponible para los nuevos clientes. Los clientes actuales de Amazon S3 Select pueden seguir utilizando la característica de la forma habitual. Más información

Amazon S3 Select admite las siguientes funciones de fecha.

Temas

- DATE_ADD
- DATE_DIFF
- EXTRACT
- TO_STRING
- TO_TIMESTAMP
- UTCNOW

DATE_ADD

Dadas una parte de fecha, una cantidad y una marca temporal, DATE_ADD devuelve una marca temporal actualizada alterando la parte de fecha por la cantidad especificada.

Sintaxis

```
DATE_ADD( date_part, quantity, timestamp )
```

Parámetros

date_part

Especifica qué parte de la fecha se debe modificar. Puede ser una de las siguientes:

- año
- mes

- día
- hora
- minuto
- segundos

quantity

El valor que se debe aplicar a la marca temporal actualizada. Los valores positivos de *quantity* se agregan a la parte de fecha de la marca temporal, y los valores negativos se restan.

timestamp

La marca temporal de destino en la que opera la función.

Ejemplos

```
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

Dadas una parte de fecha y dos marcas temporales válidas, DATE_DIFF devuelve la diferencia entre las partes de fecha. El valor devuelto es un número entero negativo si el valor *date_part* de *timestamp1* es mayor que el valor *date_part* de *timestamp2*. El valor devuelto es un número entero positivo si el valor *date_part* de *timestamp1* es menor que el valor *date_part* de *timestamp2*.

Sintaxis

```
DATE_DIFF( date_part, timestamp1, timestamp2 )
```

Parámetros

date_part

Especifica qué parte de las marcas temporales se debe comparar. Para ver la definición de `date_part`, consulte DATE_ADD.

timestamp1

La primera marca temporal que se va a comparar.

timestamp2

La segunda marca temporal que se va a comparar.

Ejemplos

```
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

Dadas una parte de fecha y una marca temporal, `EXTRACT` devuelve el valor de la parte de fecha de la marca temporal.

Sintaxis

```
EXTRACT( date_part FROM timestamp )
```

Parámetros

date_part

Especifica qué parte de las marcas temporales se va a extraer. Puede ser una de las siguientes:

- `YEAR`
- `MONTH`

- DAY
- HOUR
- MINUTE
- SECOND
- TIMEZONE_HOUR
- TIMEZONE_MINUTE

timestamp

La marca temporal de destino en la que opera la función.

Ejemplos

```
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

Dados una marca temporal y un patrón de formato, TO_STRING devuelve una representación de cadena de la marca temporal en el formato especificado.

Sintaxis

```
TO_STRING ( timestamp time_format_pattern )
```

Parámetros

timestamp

La marca temporal de destino en la que opera la función.

time_format_pattern

Una cadena que tiene las siguientes interpretaciones de caracteres especiales:

Formato	Ejemplo	Descripción
yy	69	Año en 2 dígitos
y	1969	Año en 4 dígitos
yyyy	1969	Año en 4 dígitos rellenado con ceros
M	1	Mes del año
MM	01	Mes del año rellenado con ceros
MMM	Jan	Nombre del mes del año abreviado
MMMM	January	Nombre del mes del año completo
MMMMM	J	Primera letra del mes del año (NOTA: este formato no es válido para su uso con la función TO_TIMESTAMP).

Formato	Ejemplo	Descripción
d	2	Día del mes (1-31)
dd	02	Día del mes rellenado con ceros (01-31)
a	AM	AM o PM
h	3	Hora del día (1-12)
hh	03	Hora del día rellenada con ceros (01-12)
H	3	Hora del día (0-23)
HH	03	Hora del día rellenada con ceros (00-23)
m	4	Minuto (0-59)
mm	04	Minutos rellenados con ceros (00-59)
s	5	Segundo (0-59)
ss	05	Segundos rellenados con ceros (00-59)

Formato	Ejemplo	Descripción
S	0	Fracción de un segundo (precisión: 0,1, rango: 0,0-0,9)
SS	6	Fracción de un segundo (precisión: 0,01, rango: 0,0-0,99)
SSS	60	Fracción de un segundo (precisión: 0,001, rango: 0,0-0,999)
…	…	…
SSSSSSSSS	60000000	Fracción de un segundo (precisión máxima: 1 nanosegundo, rango: 0,0-0,999 999999)
n	60000000	Nanosegundo
X	+07 o Z	Desplazamiento en horas o Z si el desplazamiento es 0

Formato	Ejemplo	Descripción
XX o XXXX	+0700 o Z	Desplazamiento en horas y minutos o Z si el desplazamiento es 0
XXX o XXXXX	+07:00 o Z	Desplazamiento en horas y minutos o Z si el desplazamiento es 0
x	7	Desplazamiento en horas
xx o xxxx	700	Desplazamiento en horas y minutos
xxx o xxxxx	+07:00	Desplazamiento en horas y minutos

Ejemplos

```
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

Dada una cadena, `TO_TIMESTAMP` la convierte en una marca temporal. `TO_TIMESTAMP` es la operación inversa de `TO_STRING`.

Sintaxis

```
TO_TIMESTAMP ( string )
```

Parámetros

`string`

La cadena de destino sobre la que opera la función.

Ejemplos

```
TO_TIMESTAMP('2007T')                          -- `2007T`
TO_TIMESTAMP('2007-02-23T12:14:33.079-08:00') -- `2007-02-23T12:14:33.079-08:00`
```

UTCNOW

`UTCNOW` devuelve la hora actual en UTC como una marca temporal.

Sintaxis

```
UTCNOW()
```

Parámetros

`UTCNOW` no acepta parámetros.

Ejemplos

```
UTCNOW() -- 2017-10-13T16:02:11.123Z
```

Funciones de cadena

> **Important**
>
> Amazon S3 Select ya no está disponible para los nuevos clientes. Los clientes actuales de Amazon S3 Select pueden seguir utilizando la característica de la forma habitual. Más información

Amazon S3 Select admite las siguientes funciones de cadena.

Temas

- CHAR_LENGTH, CHARACTER_LENGTH
- LOWER
- SUBSTRING
- TRIM
- UPPER

CHAR_LENGTH, CHARACTER_LENGTH

`CHAR_LENGTH` (o `CHARACTER_LENGTH`) cuenta el número de caracteres de la cadena especificada.

> **Note**
>
> `CHAR_LENGTH` y `CHARACTER_LENGTH` son sinónimos.

Sintaxis

```
CHAR_LENGTH ( string )
```

Parámetros

`string`

La cadena de destino sobre la que opera la función.

Ejemplos

```
CHAR_LENGTH('')          -- 0
CHAR_LENGTH('abcdefg')   -- 7
```

LOWER

Dada una cadena, `LOWER` convierte todos los caracteres en mayúsculas a minúsculas. Los caracteres que no estén en mayúscula permanecen igual.

Sintaxis

```
LOWER ( string )
```

Parámetros

string

La cadena de destino sobre la que opera la función.

Ejemplos

```
LOWER('AbCdEfG!@#$') -- 'abcdefg!@#$'
```

SUBSTRING

Dada una cadena, un índice de comienzo y, de forma opcional, una longitud, `SUBSTRING` devuelve la subcadena que va desde el índice de comienzo hasta el final de la cadena, o hasta la longitud especificada.

> **Note**
>
> El primer carácter de la cadena de entrada tiene una posición de índice de 1.
>
> - Si `start` es < 1, sin especificar una longitud, la posición de índice se establece a 1.
> - Si `start` es < 1, con la longitud especificada, la posición de índice se establece a `start + length -1`.
> - Si `start + length -1` < 0, se devuelve una cadena vacía.

- Si `start + length -1 > = 0`, se devuelve la subcadena que comienza en la posición de índice 1 con la longitud `start + length - 1`.

Sintaxis

```
SUBSTRING( string FROM start [ FOR length ] )
```

Parámetros

string

La cadena de destino sobre la que opera la función.

start

La posición de inicio de la cadena.

length

La longitud de la subcadena que se va a devolver. Si no existe, continúa hasta el final de la cadena.

Ejemplos

```
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

Elimina los caracteres anteriores o posteriores de una cadena. De forma predeterminada, el carácter que se elimina es un espacio (`' '`).

Sintaxis

```
TRIM ( [[LEADING | TRAILING | BOTH remove_chars] FROM] string )
```

Parámetros

string

La cadena de destino sobre la que opera la función.

LEADING | TRAILING | BOTH

Este parámetro indica si se deben eliminar los caracteres anteriores o posteriores, o ambos.

remove_chars

El conjunto de caracteres que se debe eliminar. *remove_chars* puede ser una cadena con longitud > 1. Esta función devuelve la cadena con los caracteres de *remove_chars* encontrados al principio o al final de la cadena que se han eliminado.

Ejemplos

```
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

Dada una cadena, UPPER convierte todos los caracteres en minúsculas a mayúsculas. Los caracteres que no estén en minúscula permanecen igual.

Sintaxis

```
UPPER ( string )
```

Parámetros

string

La cadena de destino sobre la que opera la función.

Ejemplos

```
UPPER('AbCdEfG!@#$') -- 'ABCDEFG!@#$'
```

Realización de operaciones por lotes a gran escala en objetos de Amazon S3

Puede utilizar Operaciones por lotes de S3 para realizar operaciones por lotes a gran escala en objetos de Amazon S3. Operaciones por lotes de S3 puede realizar una sola operación en las listas de objetos de Amazon S3 que especifique. Un solo trabajo puede realizar una operación especificada en miles de millones de objetos que contiene exabytes de datos. Amazon S3 realiza un seguimiento del avance, envía notificaciones y guarda un informe de finalización de todas las acciones, por lo que proporciona una experiencia sin servidor, auditable y completamente administrada. Puede emplear la herramienta de operaciones por lotes de S3 a través de la AWS Management Console, la AWS CLI, los SDK de Amazon o la API REST.

Utilice Operaciones por lotes de S3 para copiar objetos y establecer etiquetas de objetos o listas de control de acceso (ACL). También puede iniciar restauraciones de objetos desde S3 Glacier Flexible Retrieval o invocar una función de AWS Lambda que realice acciones personalizadas con los objetos. Puede realizar estas operaciones en una lista personalizada de objetos o puede utilizar un informe de Amazon S3 Inventory para generar listas de objetos fácilmente. Las operaciones por lotes de Amazon S3 utilizan las mismas API de Amazon S3 que ya utiliza con Amazon S3, por lo que la interfaz le resultará familiar.

> **Note**
>
> Para obtener más información sobre el uso de la clase de almacenamiento Amazon S3 Express One Zone con buckets de directorio, consulte ¿Qué es S3 Express One Zone? y Buckets de directorio. Para obtener más información sobre el uso de Operaciones por lotes con S3 Express One Zone y buckets de directorio, consulte Uso de operaciones por lotes con S3 Express One Zone.

Conceptos básicos de Operaciones por lotes de S3

Puede utilizar Operaciones por lotes de S3 para realizar operaciones por lotes a gran escala en objetos de Amazon S3. Las operaciones por lotes de S3 pueden ejecutar una sola operación en las listas de objetos de Amazon S3 que especifique.

Terminología

En esta sección, se utilizan los términos trabajo, operación y tarea, cuyas definiciones son las siguientes:

Trabajo

Un trabajo es la unidad básica de trabajo para las operaciones por lotes de S3. Un trabajo contiene toda la información necesaria para ejecutar la operación especificada en los objetos enumerados en el manifiesto. Una vez que se proporciona esta información y se solicita que el trabajo comience, el trabajo realiza la operación en cada objeto del manifiesto.

Operation

La operación es el tipo de acción de API, como copiar objetos, que desea que ejecute el trabajo de Operaciones por lotes. Cada trabajo realiza un único tipo de operación en todos los objetos especificados en el manifiesto.

Tarea

Una tarea es la unidad de ejecución de un trabajo. Una tarea representa una llamada única a una operación de API de Amazon S3 o AWS Lambda para realizar la operación del trabajo en un único objeto. En el transcurso de la vida útil de un trabajo, Operaciones por lotes de S3 crea una tarea para cada objeto especificado en el manifiesto.

Cómo funciona un trabajo de operaciones por lotes de S3

Un trabajo es la unidad básica de trabajo para las operaciones por lotes de S3. Un trabajo contiene toda la información necesaria para ejecutar la operación especificada en una lista de objetos. Para crear un trabajo, debe proporcionar a las operaciones por lotes de S3 una lista de objetos y especificar la acción que se debe realizar con dichos objetos.

Para obtener información acerca de las operaciones que admiten las operaciones por lotes de S3, consulte Operaciones compatibles con las operaciones por lotes de S3.

Los trabajos por lotes realizan la operación especificada en cada uno de los objetos incluidos en su manifiesto. Un manifiesto enumera los objetos que desea que procese un trabajo por lotes y se almacena como un objeto en un bucket. Puede utilizar un informe de Inventario de Amazon S3 con formato CSV (valores separados por comas) como manifiesto, lo que facilita la creación de grandes listas de objetos ubicados en un bucket. También puede especificar un manifiesto en un formato CSV sencillo que le permite realizar operaciones por lotes en una lista personalizada de objetos incluidos en un solo bucket.

Después de crear un trabajo, Amazon S3 procesa la lista de objetos del manifiesto y ejecuta la operación especificada en cada objeto. Mientras el trabajo se ejecuta, puede monitorear su avance desde un programa o a través de la consola de Amazon S3. También puede configurar un trabajo para generar un informe de finalización cuando haya terminado. En el informe de finalización, se describen los resultados de cada una de las tareas realizadas por el trabajo. Para obtener más información sobre el monitoreo de trabajos, consulte Administración de trabajos de operaciones por lotes de S3.

Tutorial operaciones por lotes de S3

En el siguiente tutorial se presentan procedimientos integrales completos para algunas tareas de operaciones por lotes.

- Tutorial: videos de transcodificación por lotes con operaciones por lotes de S3, AWS Lambda, y AWS Elemental MediaConvert

Concesión de permisos para Operaciones por lotes de S3

Antes de crear y ejecutar trabajos de operaciones por lotes de S3, debe conceder los permisos necesarios. Para crear un trabajo de operaciones por lotes de Amazon S3 se requiere el permiso del usuario `s3:CreateJob`. La misma entidad que crea el trabajo también debe tener permiso `iam:PassRole` para transferir el rol de AWS Identity and Access Management (IAM) especificado para el trabajo a la herramienta de operaciones por lotes.

Para obtener información general acerca de cómo especificar recursos de IAM, consulte Elementos de la política de JSON de IAM: Resource en la Guía del usuario de IAM. En las siguientes secciones se proporciona información sobre cómo crear un rol de IAM y cómo adjuntar las políticas.

Temas

- Creación de un rol de IAM de Operaciones por lotes de S3

- Adjunción de políticas de permisos

Creación de un rol de IAM de Operaciones por lotes de S3

Amazon S3 debe tener permisos para poder realizar operaciones por lotes de S3 en su nombre. Estos permisos se conceden a través de un rol de AWS Identity and Access Management (IAM). Esta sección proporciona ejemplos de las políticas de confianza y de permisos que se utilizan al crear un rol de IAM. Para obtener más información, consulte Roles de IAM en la Guía del usuario de IAM. Para ver ejemplos, consulte Control de permisos para Operaciones por lotes de S3 mediante etiquetas de trabajo y Copia de objetos mediante operaciones por lotes de S3.

En las políticas de IAM también puede utilizar claves de condición para filtrar permisos de acceso para trabajos de Operaciones por lotes de S3. Para obtener más información y una lista completa de las claves de condición específicas de Amazon S3, consulte Actions, resources, and condition keys for Amazon S3 en la Referencia de autorización de servicios.

Política de confianza

Para que la entidad principal del servicio Operaciones por lotes de S3 pueda asumir el rol de IAM debe asociar al rol la siguiente política de confianza.

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

Adjunción de políticas de permisos

En función del tipo de operaciones, puede asociar una de estas políticas.

Antes de configurar los permisos, tenga en cuenta lo siguiente:

- Con independencia de la operación que realice, Amazon S3 necesita permisos para leer el objeto del manifiesto del bucket de S3 y, de forma opcional, escribir un informe en el bucket. Por lo tanto, todas las políticas que se indican a continuación contienen estos permisos.
- Para los manifiestos de informes de Amazon S3 Inventory, S3 Batch Operations requiere permiso para leer el objeto manifest.json y todos los archivos de datos CSV asociados.
- Los permisos específicos de las versiones como `s3:GetObjectVersion` solo son obligatorios cuando especifica el ID de la versión de los objetos.
- Si ejecuta la herramienta de operaciones por lotes de S3 en objetos cifrados, el rol de IAM también debe tener acceso a las claves de AWS KMS utilizadas para cifrarlos.
- Si envía un manifiesto de informe de inventario cifrado con AWS KMS, su política de IAM debe incluir los permisos `"kms:Decrypt"` y `"kms:GenerateDataKey"` para el objeto manifest.json y todos los archivos de datos CSV asociados.
- Si el trabajo de Operaciones por lote genera un manifiesto en un bucket que tiene las ACL habilitadas y se encuentra en una cuenta de AWS diferente, debe conceder el permiso `s3:PutObjectAcl` en la política de IAM del rol de IAM configurado para el trabajo por lotes. Si no incluye este permiso, el trabajo por lotes no se realiza y genera el error `Error occurred when preparing manifest: Failed to write manifest`.

Copiar objetos: 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/*"
            ]
        }
    ]
}
```

Reemplazar el etiquetado de objetos: 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/*"
      ]
    }
  ]
}
```

Eliminar etiquetado de objetos: 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/*"
            ]
        }
    ]
}
```

Reemplazar lista de control de acceso: 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/*"
      ]
    }
  ]
}
```

Restaurar objetos: 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/*"
      ]
    }
  ]
}
```

Aplicar retención de bloqueo de objetos: 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/*"
            ]
        }
    ]
}
```

Aplicar bloqueo de objeto retención legal: 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/*"
            ]
        }
    ]
}
```

Replicación de objetos existentes: InitiateReplication con un manifiesto generado por S3

Utilice esta política si utiliza y almacena un manifiesto generado por S3. Para obtener más información sobre cómo utilizar las operaciones por lotes para replicar objetos existentes, consulte Replicación de objetos existentes con replicación por lotes de 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 ****/*"
         ]
      }
   ]
}
```

Replicar objetos existentes: InitiateReplication con un manifiesto del usuario

Utilice esta política si utiliza un manifiesto que proporciona el usuario. Para obtener más información sobre cómo utilizar las operaciones por lotes para replicar objetos existentes, consulte Replicación de objetos existentes con replicación por lotes de 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 ****/*"
```

```
        ]
      }
   ]
}
```

Creación de trabajos de operaciones por lotes de S3

Las Operaciones por lotes de Amazon S3 le permiten realizar operaciones por lotes a gran escala en una lista de objetos de Amazon S3 concretos. Esta sección contiene la información que necesita para crear un trabajo de Operaciones por lotes de S3 y los resultados de una solicitud `CreateJob`. También proporciona instrucciones para crear un trabajo de la herramienta de operaciones por lotes con la consola, AWS Command Line Interface (AWS CLI) y AWS SDK for Java.

Cuando crea un trabajo de operaciones por lotes de S3, puede solicitar un reporte de finalización para todas las tareas o solo para las tareas que no se realicen. Siempre que se haya invocado correctamente al menos una tarea, Operaciones por lotes de S3 genera un informe para los trabajos que se han completado, que han fallado o que se han cancelado. Para obtener más información, consulte Ejemplos: informes de finalización de las operaciones por lotes de S3.

Temas

Elementos de una solicitud de trabajo de Operaciones por lotes

Para crear un trabajo de Operaciones por lotes de S3 debe proporcionar la siguiente información:

Operation

Especifique la operación que quiere que Operaciones por lotes de S3 ejecute en los objetos del manifiesto. Cada tipo de operación acepta parámetros específicos de dicha operación. Con Operaciones por lotes, puede realizar una operación por lotes con los mismos resultados que si realizase esa operación una por una en cada proyecto.

Manifiesto

El manifiesto es una lista de todos los objetos en los que quiere que Operaciones por lotes de S3 ejecute la operación especificada. Puede usar los siguientes métodos para especificar un manifiesto para un trabajo de Operaciones por lotes:

- Crear manualmente su propia lista de objetos personalizada con formato CSV.
- Elegir un informe Inventario de Amazon S3 existente con formato CSV.
- Dirigir Operaciones por lotes para generar un manifiesto automáticamente en función de los criterios de filtro de objetos que especifique al crear su trabajo. Esta opción está disponible para trabajos de replicación por lotes que cree en la consola de Amazon S3 o para cualquier tipo de trabajo que cree mediante la AWS CLI, los SDK de AWS o la API de REST de Amazon S3.

> **Note**
>
> - Independientemente de cómo especifique el manifiesto, la propia lista debe almacenarse en un bucket de uso general. Operaciones por lotes no puede importar los manifiestos existentes ni guardar los manifiestos generados en buckets de directorio. Sin embargo, los objetos descritos en el manifiesto se pueden almacenar en buckets de directorio. Para obtener más información, consulte Buckets de directorio.
> - Si los objetos del manifiesto están en un bucket versionado, especificar los ID de versión de los objetos dirige las Operaciones por lotes para realizar la operación en una versión específica. Si no se especifican los ID de versión, Operaciones por lotes realiza la operación en la versión más reciente de los objetos. Si el manifiesto incluye un campo de ID de versión, debe proporcionar un ID de versión para todos los objetos del manifiesto.

Para obtener más información, consulte Especificar un manifiesto.

Prioridad

Utilice las prioridades de los trabajos para indicar la prioridad relativa de este trabajo con respecto a otros que se estén ejecutando en la cuenta. Cuanto más elevado sea el número, mayor será la prioridad.

Las prioridades de trabajo solo tienen significado en relación con las prioridades establecidas para otros trabajos de la misma cuenta y región. Puede elegir el sistema de numeración que mejor le convenga. Por ejemplo, es posible que desee asignar una prioridad de 1 todos los trabajos Restaurar (`RestoreObject`), una prioridad de 2 a todos los trabajos Copiar (`CopyObject`) y una prioridad de 3 a todos los trabajos Reemplazar listas de control de acceso (ACL) (`PutObjectAcl`)

Operaciones por lotes de S3 establece la prioridad de los trabajos en función de los números asignados, pero el orden no siempre se mantiene de forma estricta. De este modo, no utilice

prioridades de trabajo para garantizar que cualquier trabajo comience o finalice antes que cualquier otro trabajo. Si debe garantizar un orden estricto, debe esperar hasta que un trabajo finalice para comenzar el siguiente.

RoleArn

Especifique un rol de AWS Identity and Access Management (IAM) que ejecute el trabajo. El rol de IAM que utilice debe tener permisos suficientes para realizar la operación especificada en el trabajo. Por ejemplo, para ejecutar un trabajo `CopyObject`, el rol de IAM debe tener el permiso `s3:GetObject` en el bucket de origen y el permiso `s3:PutObject` en el bucket de destino. El rol también necesita permisos para leer el manifiesto y escribir el informe de finalización del trabajo.

Para obtener más información acerca de los roles de IAM, consulte Roles de IAM en Guía del usuario de IAM.

Para obtener más información sobre los permisos de Amazon S3, consulte Acciones de políticas para Amazon S3.

> **Note**
>
> Los trabajos de Operaciones por lotes que realizan acciones en buckets de directorio requieren permisos específicos. Para obtener más información, consulte AWS Identity and Access Management (IAM) para S3 Express One Zone.

Informar

Especifique si desea que Operaciones por lotes de S3 genere un informe de finalización. Si solicita un informe de finalización del trabajo, tiene que proporcionar los parámetros del informe en este elemento. La información necesaria incluye lo siguiente:

- el bucket donde desea almacenar el reporte,

  > **Note**
  >
  > El informe debe almacenarse en un bucket de uso general. Operaciones por lotes no puede guardar los informes en los buckets de directorio. Para obtener más información, consulte Buckets de directorio.

- el formato del reporte,

- si desea que el reporte incluya los detalles de todas las tareas o solo las tareas con error,
- una cadena de prefijo opcional.

> **Note**
>
> Los informes de finalización siempre se cifran con claves administradas por Amazon S3 (SSE-S3).

Tags (Etiquetas) (opcionales)

Para etiquetar y controlar el acceso a los trabajos de Operaciones por lotes de S3, puede añadir etiquetas. Puede utilizar etiquetas para identificar quien es el responsable de un trabajo de Operaciones por lotes, o bien para controlar cuántos usuarios interactúan con los trabajos de Operaciones por lotes. La presencia de etiquetas de trabajo puede conceder o limitar la capacidad de un usuario para cancelar un trabajo, activar un trabajo en estado de confirmación o cambiar el nivel de prioridad de un trabajo. Por ejemplo, puede conceder permiso a un usuario para invocar la operación `CreateJob` siempre que el trabajo se haya creado con la etiqueta `"Department=Finance"`.

Puede crear trabajos con etiquetas asociadas a ellos y puede añadir etiquetas a los trabajos después de crearlos.

Para obtener más información, consulte the section called “Uso de etiquetas”.

Descripción (opcional)

Para realizar un seguimiento y monitorear su trabajo también puede proporcionar una descripción de hasta 256 caracteres. Amazon S3 incluye esta descripción siempre que devuelve información sobre un trabajo o muestra los detalles del trabajo en la consola de Amazon S3. Puede ordenar y filtrar los trabajos fácilmente en función de las descripciones asignadas. Las descripciones no tienen que ser únicas, por lo que puede usar descripciones como categorías (por ejemplo: "trabajos de copia de registros semanales") que le ayuden a hacer un seguimiento de los grupos de trabajos parecidos.

Especificar un manifiesto

Un manifiesto es un objeto de Amazon S3 que contiene las claves de objeto sobre las que desea que actúe Amazon S3. Puede proporcionar un manifiesto de una de las siguientes formas:

- Crear un nuevo archivo de manifiesto manualmente.
- Usar un manifiesto existente.
- Dirigir Operaciones por lotes para generar un manifiesto automáticamente en función de los criterios de filtro de objetos que especifique al crear su trabajo. Esta opción está disponible para trabajos de replicación por lotes que cree en la consola de Amazon S3 o para cualquier tipo de trabajo que cree mediante la AWS CLI, los SDK de AWS o la API de REST de Amazon S3.

> **Note**
>
> Independientemente de cómo especifique el manifiesto, la propia lista debe almacenarse en un bucket de uso general. Operaciones por lotes no puede importar los manifiestos existentes ni guardar los manifiestos generados en buckets de directorio. Sin embargo, los objetos descritos en el manifiesto se pueden almacenar en buckets de directorio. Para obtener más información, consulte Buckets de directorio.

Creación de un archivo de manifiesto

Si crea un archivo manifiesto manualmente, debe especificar la clave de objeto del manifiesto, una ETag (etiqueta de entidad) y un ID de versión opcional en una lista con formato CSV. El contenido del manifiesto debe tener codificación URL.

De forma predeterminada, Amazon S3 utiliza automáticamente el cifrado del servidor con claves administradas de Amazon S3 (SSE-S3) para cifrar un manifiesto que se carga en un bucket de Amazon S3. No se admiten los manifiestos que utilizan cifrado del lado del servidor con claves proporcionadas por el cliente (SSE-C). Solo se admiten los manifiestos que utilizan el cifrado del servidor con claves de AWS Key Management Service (AWS KMS) (SSE-KMS) cuando se utilizan informes de inventario con formato CSV. No se admite el uso de un manifiesto creado manualmente con AWS KMS.

El manifiesto debe contener el nombre del bucket, la clave del objeto y, de manera opcional, la versión del objeto. Las operaciones por lotes de S3 no utilizan ninguno de los demás campos.

> **Note**
>
> Si los objetos del manifiesto están en un bucket versionado, especificar los ID de versión de los objetos dirige las Operaciones por lotes para realizar la operación en una versión

específica. Si no se especifican los ID de versión, Operaciones por lotes realiza la operación en la versión más reciente de los objetos. Si el manifiesto incluye un campo de ID de versión, debe proporcionar un ID de versión para todos los objetos del manifiesto.

A continuación, se muestra un ejemplo de un manifiesto en formato CSV sin los ID de versión.

```
Examplebucket,objectkey1
Examplebucket,objectkey2
Examplebucket,objectkey3
Examplebucket,photos/jpgs/objectkey4
Examplebucket,photos/jpgs/newjersey/objectkey5
Examplebucket,object%20key%20with%20spaces
```

A continuación, se muestra un manifiesto de ejemplo en formato CSV que incluye los ID de versión.

```
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
```

Especificación de un archivo de manifiesto existente

Puede especificar un archivo de manifiesto en una solicitud de creación de trabajo utilizando uno de los dos formatos siguientes.

- Informe de Inventario de Amazon S3: debe ser un informe de Inventario de Amazon S3 con formato CSV. Debe especificar el archivo `manifest.json` que está asociado con el informe de inventario. Para obtener más información sobre los informes de inventario, consulte Inventario de Amazon S3. Si el informe de inventario incluye los identificadores de las versiones, las Operaciones por lotes de S3 se ejecutan en las versiones del objeto especificadas.

  > **Note**
  >
  > - Las operaciones por lotes de S3 son compatibles con los informes de inventario que están cifrados con SSE-KMS.

- Si envía un manifiesto de informe de inventario cifrado con SSE-KMS, su política de IAM debe incluir los permisos `"kms:GenerateDataKey"` y `manifest.json` para el objeto `"kms:Decrypt"` y todos los archivos de datos CSV asociados.

- Archivo CSV: cada fila del archivo debe incluir el nombre del bucket, la clave del objeto y, de manera opcional, la versión del objeto. Las claves de objeto deben estar codificadas como URL, tal y como se muestra en los siguientes ejemplos. El manifiesto debe incluir los ID de versión de todos los objetos u omitirlos. Para obtener más información acerca de el formato de manifiesto de CSV, consulte JobManifestSpec en la Referencia de la API de Amazon Simple Storage Service.

> **Note**
>
> Las Operaciones por lotes de S3 no admiten archivos de manifiesto CSV cifrados por SSE-KMS.

> **Important**
>
> Cuando utiliza un manifiesto creado manualmente y un bucket con control de versiones, le recomendamos que especifique los ID de versión de los objetos. Cuando se crea un objeto, las Operaciones por lotes de S3 analizan todo el manifiesto antes de ejecutar el trabajo. Sin embargo, no realizan una "instantánea" del estado del bucket.
> Puesto que los manifiestos pueden contener miles de millones de objetos, los trabajos tardan mucho tiempo en ejecutarse, lo que puede afectar a la versión de un objeto sobre la que actúa el trabajo. Supongamos que sobrescribe un objeto con una versión nueva mientras un trabajo está en ejecución y no especificó el ID de versión de ese objeto. En este caso, Amazon S3 realiza la operación en la última versión del objeto, no en la versión que existe cuando se creó el trabajo. La única manera de evitar este comportamiento consiste en especificar los ID de versión de los objetos que aparecen en el manifiesto.

Generación automática de un manifiesto

Puede indicar a Amazon S3 que genere un manifiesto automáticamente en función de los criterios de filtro de objetos que especifique al crear su trabajo. Esta opción está disponible para trabajos de replicación por lotes que cree en la consola de Amazon S3 o para cualquier tipo de trabajo que cree mediante la AWS CLI, los SDK de AWS o la API de REST de Amazon S3. Para obtener más

información sobre la replicación por lotes, consulte Replicación de objetos existentes con replicación por lotes de S3.

Para generar un manifiesto automáticamente, debe especificar los siguientes elementos como parte de su solicitud de creación de trabajo:

- Información sobre el bucket que contiene los objetos de origen, incluidos el propietario del bucket y el nombre de recurso de Amazon (ARN)
- Información sobre la salida del manifiesto, incluida una marca para crear un archivo de manifiesto, el propietario del bucket de salida, el ARN, el prefijo, el formato de archivo y el tipo de cifrado
- Criterios opcionales para filtrar los objetos por fecha de creación, nombre de clave, tamaño, clase de almacenamiento y etiquetas

Criterios de filtro de objetos

Para filtrar la lista de objetos que se van a incluir en un manifiesto generado automáticamente, puede especificar los siguientes criterios. Para obtener más información, consulte JobManifestGeneratorFilter en la Referencia de la API de Amazon S3.

CreatedAfter

Si se proporciona, el manifiesto generado incluye solo los objetos del bucket de origen que se crearon después de este tiempo.

CreatedBefore

Si se proporciona, el manifiesto generado incluye solo los objetos del bucket de origen que se crearon antes de este tiempo.

EligibleForReplication

Si se proporciona, el manifiesto generado incluye los objetos solo si son aptos para la replicación de acuerdo con la configuración de replicación del bucket de origen.

KeyNameConstraint

Si se proporciona, el manifiesto generado incluye solo los objetos del bucket de origen cuyas claves de objeto coincidan con las restricciones de cadena especificadas para MatchAnySubstring, MatchAnyPrefix, and MatchAnySuffix

MatchAnySubstring si se proporciona, el manifiesto generado incluye objetos si la cadena especificada aparece en cualquier parte de la cadena de la clave del objeto.

MatchAnyPrefix: si se proporciona, el manifiesto generado incluye objetos si la cadena especificada aparece al principio de la cadena de la clave del objeto.

MatchAnySuffix: si se proporciona, el manifiesto generado incluye objetos si la cadena especificada aparece al final de la cadena de la clave del objeto.

MatchAnyStorageClass

Si se proporciona, el manifiesto generado incluye solo los objetos del bucket de origen que se almacenan con la clase de almacenamiento especificada.

ObjectReplicationStatuses

Si se proporciona, el manifiesto generado incluye solo los objetos del bucket de origen que tengan uno de los estados de replicación especificados.

ObjectSizeGreaterThanBytes

Si se proporciona, el manifiesto generado incluye solo los objetos del bucket de origen cuyo tamaño de archivo sea superior al número de bytes especificado.

ObjectSizeLessThanBytes

Si se proporciona, el manifiesto generado incluye solo los objetos del bucket de origen cuyo tamaño de archivo sea inferior al número de bytes especificado.

> **Note**
>
> No puede clonar la mayoría de los trabajos que tienen manifiestos generados automáticamente. Los trabajos de replicación por lotes se pueden clonar, excepto cuando utilizan los criterios de filtro de manifiesto `KeyNameConstraint`, `MatchAnyStorageClass`, `ObjectSizeGreaterThanBytes` o `ObjectSizeLessThanBytes`.

La sintaxis para especificar los criterios del manifiesto varía en función del método que utilice para crear el trabajo. Para ver ejemplos, consulte Creación de un trabajo.

Creación de un trabajo

Puede crear trabajos de operaciones por lotes de S3 mediante la consola de Amazon S3, AWS CLI, los SDK de AWS o la API de REST de Amazon S3.

Para obtener más información acerca de la creación de una solicitud de trabajo, consulte Elementos de una solicitud de trabajo de Operaciones por lotes.

Requisitos previos

Antes de crear un trabajo de operaciones por lotes, confirme que ha configurado los permisos pertinentes. Para obtener más información, consulte Concesión de permisos para Operaciones por lotes de S3.

Uso de la consola de S3

Para crear un trabajo por lotes

1. Inicie sesión AWS Management Console Management Console y abra la consola de Amazon S3 en https://console.aws.amazon.com/s3/.
2. En la barra de navegación de la parte superior de la página, elija el nombre de la Región de AWS que aparece. A continuación, elija en Region (Región) la región en la que desea crear el trabajo.

   > **Note**
   >
   > Para las operaciones de copia, debe crear el trabajo en la misma región que el bucket de destino. Para las demás operaciones, debe crear el trabajo en la misma región que los objetos en el manifiesto.

3. Seleccione Batch Operations (Operaciones por lotes) en el panel de navegación izquierdo de la consola de Amazon S3.
4. Seleccione Crear trabajo.
5. Vea la Región de AWS en la que desea crear el trabajo.
6. En Manifest format (Formato del manifiesto), seleccione el tipo de objeto del manifiesto que desee usar.

   - Si elige S3 Inventory report (Informe de inventario de S3), escriba la ruta del objeto manifest.json que Amazon S3 ha generado con el informe de inventario con formato CSV. También puede seleccionar el ID de versión del objeto del manifiesto en caso de que desee utilizar otra versión que no sea la más reciente.
   - Si selecciona CSV, escriba la ruta del objeto del manifiesto con formato CSV. El objeto del manifiesto debe tener el mismo formato que se ha especificado en la consola. Si quiere

utilizar otra versión que no sea la más reciente, puede incluir el ID de versión del objeto del manifiesto.

> **Note**
>
> La consola de Amazon S3 solo admite la generación automática de manifiestos para trabajos de replicación por lotes. Para el resto de tipos de trabajo, si desea que Amazon S3 genere un manifiesto automáticamente en función de los criterios de filtro que especifique, debe configurar el trabajo mediante la AWS CLI, los SDK de AWS o la API de REST de Amazon S3.

7. Elija Siguiente.
8. En Operation (Operación), seleccione la operación que desee en todos los objetos que aparecen en el manifiesto. Rellene los datos de la operación seleccionada y haga clic en Siguiente.
9. Rellene los datos de Configurar otras opciones y haga clic en Siguiente.
10. En Review (Revisar), compruebe la configuración. Si necesita realizar cambios, seleccione Anterior. De lo contrario, seleccione Crear trabajo.

Uso de la AWS CLI

Specify manifest

En el siguiente ejemplo se muestra cómo crea un trabajo `S3PutObjectTagging` de Operaciones por lotes de S3 que actúa en objetos que se enumeran en un archivo de manifiesto existente.

Para crear un trabajo **`S3PutObjectTagging`** de Operaciones por lotes, realice las siguientes acciones:

1. Utilice los siguientes comandos para crear un rol de AWS Identity and Access Management (IAM) y, a continuación, cree una política de IAM para asignar los permisos pertinentes. El siguiente rol y política conceden permiso a Amazon S3 para añadir etiquetas de objetos, que serán necesarias al crear el trabajo en un paso posterior.

   a. Utilice el siguiente comando de ejemplo para crear un rol de IAM para que lo utilicen las Operaciones por lotes. Para usar este comando de ejemplo, reemplace *`S3BatchJobRole`* por el nombre que desea asignar a esta aplicación.

```
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"
      }
   ]
}'
```

Registre el nombre de recurso de Amazon (ARN) del rol. Lo necesitará para poder crear los trabajos.

b. Utilice el siguiente comando de ejemplo para crear una política de IAM con los permisos necesarios y asóciela al rol de IAM que creó en el paso anterior. Para obtener información sobre los permisos de necesarios, consulte Concesión de permisos para Operaciones por lotes de S3.

> **Note**
>
> Los trabajos de Operaciones por lotes que realizan acciones en buckets de directorio requieren permisos específicos. Para obtener más información, consulte AWS Identity and Access Management (IAM) para S3 Express One Zone.

Para usar este comando de ejemplo, sustituya *`user input placeholders`* de la siguiente manera:

- Sustituya *`S3BatchJobRole`* por el nombre de su rol de IAM. Asegúrese de que este nombre coincide con el nombre que utilizó anteriormente.
- Sustituya *`PutObjectTaggingBatchJobPolicy`* por el nombre que desea asignar a la política de IAM.

- Sustituya *amzn-s3-demo-destination-bucket* por el nombre del bucket que contiene los objetos a los que desea aplicar etiquetas.
- Sustituya *DOC-EXAMPLE-MANIFEST-BUCKET* por el nombre del bucket de que contiene el manifiesto.
- Sustituya *DOC-EXAMPLE-REPORT-BUCKET* por el nombre del bucket donde desea que se entregue el informe de finalización.

```
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:::amzn-s3-demo-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. Utilice el siguiente comando de ejemplo para crear un trabajo `S3PutObjectTagging`.

   El archivo `manifest.csv` proporciona una lista de valores de claves de objetos y buckets. El trabajo aplica las etiquetas especificadas a los objetos que se identifican en el manifiesto. `ETag` es la etiqueta de entidad del objeto `manifest.csv`, que puede obtenerse desde la consola de Amazon S3. En esta solicitud, se especifica el parámetro `no-confirmation-required` para que pueda ejecutar el trabajo sin tener que confirmarlo con el comando `update-job-status`. Para obtener más información, consulte create-job en la Referencia de los comandos de AWS CLI.

   Para utilizar este comando de ejemplo, sustituya *`user input placeholders`* por su propia información. Sustituya *`IAM-role`* por el ARN del rol de IAM que ha creado anteriormente.

```
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
```

Como respuesta, Amazon S3 devuelve el ID del trabajo (por ejemplo: `00e123a4-c0d8-41f4-a0eb-b46f9ba5b07c`). Será necesario el ID de trabajo para identificar, supervisar y modificar el trabajo.

Generate manifest

En el siguiente ejemplo se muestra cómo crear un trabajo `S3DeleteObjectTagging` de Operaciones por lotes de S3 que genera automáticamente un manifiesto basado en los criterios del filtro de objeto. Este criterio incluye la fecha de creación, el nombre de la clave, el tamaño, la clase de almacenamiento y las etiquetas.

Para crear un trabajo **`S3DeleteObjectTagging`** de Operaciones por lotes, realice las siguientes acciones:

1. Utilice los siguientes comandos para crear un rol de AWS Identity and Access Management (IAM) y, a continuación, cree una política de IAM para asignar permisos. El siguiente rol y política conceden permiso a Amazon S3 para eliminar etiquetas de objetos, que serán necesarias al crear el trabajo en un paso posterior.

   a. Utilice el siguiente comando de ejemplo para crear un rol de IAM para que lo utilicen las Operaciones por lotes. Para usar este comando de ejemplo, reemplace *`S3BatchJobRole`* por el nombre que desea asignar a esta aplicación.

```
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"
      }
   ]
}'
```

Registre el nombre de recurso de Amazon (ARN) del rol. Lo necesitará para poder crear los trabajos.

b. Utilice el siguiente comando de ejemplo para crear una política de IAM con los permisos necesarios y asóciela al rol de IAM que creó en el paso anterior. Para obtener información sobre los permisos de necesarios, consulte Concesión de permisos para Operaciones por lotes de S3.

> **Note**
>
> Los trabajos de Operaciones por lotes que realizan acciones en buckets de directorio requieren permisos específicos. Para obtener más información, consulte AWS Identity and Access Management (IAM) para S3 Express One Zone.

Para usar este comando de ejemplo, sustituya *`user input placeholders`* de la siguiente manera:

- Sustituya *`S3BatchJobRole`* por el nombre de su rol de IAM. Asegúrese de que este nombre coincide con el nombre que utilizó anteriormente.
- Sustituya *`DeleteObjectTaggingBatchJobPolicy`* por el nombre que desea asignar a la política de IAM.
- Sustituya *`amzn-s3-demo-destination-bucket`* por el nombre del bucket que contiene los objetos a los que desea aplicar etiquetas.
- Sustituya *`DOC-EXAMPLE-MANIFEST-OUTPUT-BUCKET`* por el nombre del bucket donde desee guardar el manifiesto.
- Sustituya *`DOC-EXAMPLE-REPORT-BUCKET`* por el nombre del bucket donde desea que se entregue el informe de finalización.

```
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:::amzn-s3-demo-destination-bucket/*"
    },
    {
      "Effect":"Allow",
      "Action":[
        "s3:PutInventoryConfiguration"
      ],
      "Resource": "arn:aws:s3:::amzn-s3-demo-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. Utilice el siguiente comando de ejemplo para crear el trabajo `S3DeleteObjectTagging`.

En este ejemplo, los valores de la sección `--report` especifican el bucket, el prefijo, el formato y el alcance del informe de trabajo que se generará. La sección `--manifest-generator` especifica información sobre el bucket de origen que contiene los objetos sobre los que actuará el trabajo, información sobre la lista de resultados del manifiesto que se generará para el trabajo y los criterios de filtro para limitar el alcance de los objetos que se van a incluir en el manifiesto por fecha de creación, restricciones de nombre, tamaño y clase de almacenamiento. El comando también especifica la prioridad del trabajo, el rol de IAM y Región de AWS.

Para obtener más información, consulte create-job en la Referencia de los comandos de AWS CLI.

Para utilizar este comando de ejemplo, sustituya *user input placeholders* por su propia información. Sustituya *IAM-role* por el ARN del rol de IAM que ha creado anteriormente.

```
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
```

Como respuesta, Amazon S3 devuelve el ID del trabajo (por ejemplo: `00e123a4-c0d8-41f4-a0eb-b46f9ba5b07c`). Será necesario el ID de este trabajo para identificar, supervisar o modificar el trabajo.

Mediante AWS SDK for Java

Specify manifest

En el siguiente ejemplo se muestra cómo crea un trabajo `S3PutObjectTagging` de Operaciones por lotes de S3 que actúa en objetos que se enumeran en un archivo de manifiesto existente. Para utilizar este ejemplo, reemplace los *`user input placeholders`* con su propia información.

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

En el siguiente ejemplo se muestra cómo crear un trabajo `s3PutObjectCopy` de Operaciones por lotes de S3 que genere automáticamente un manifiesto en función de los criterios de filtrado de objetos, incluida la fecha de creación, el nombre de la clave y el tamaño. Para utilizar este ejemplo, reemplace los *`user input placeholders`* con su propia información.

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();
            }
        }
    }
```

Uso de la API de REST

Puede utilizar la API de REST para crear un trabajo de operaciones por lotes. Para obtener más información, consulte CreateJob en la Referencia de la API de Amazon Simple Storage Service.

Respuestas del trabajo

Si la solicitud `CreateJob` se realiza correctamente, Amazon S3 devuelve un ID de trabajo. El ID de trabajo es un identificador único que Amazon S3 genera automáticamente para que pueda identificar el trabajo de operaciones por lotes y monitorear su estado.

Cuando cree un trabajo a través de la AWS CLI, los SDK de AWS o la API de REST, puede definir Operaciones por lotes de S3 para comenzar a procesar el trabajo automáticamente. El trabajo se ejecuta en cuanto está listo en lugar de esperar a que se ejecuten otros trabajos de mayor prioridad.

Cuando cree un trabajo a través de la consola de Amazon S3, tendrá que revisar los detalles del trabajo y confirmar que desea ejecutarlo para que la herramienta de operaciones por lotes

pueda comenzar a procesarlo. Si un trabajo permanece suspendido durante más de 30 días, no se ejecutará.

Operaciones compatibles con las operaciones por lotes de S3

Operaciones por lotes de S3 admite varias operaciones diferentes. En los temas de esta sección, se describen las distintas operaciones.

Copia de objetos

Mediante la operación Copy, se copia cada objeto especificado en el manifiesto. Puede copiar objetos en un bucket en la misma región de AWS o en un bucket de una región diferente. Operaciones por lotes de S3 admite la mayoría de las opciones disponibles a través de Amazon S3 para copiar objetos. Estas opciones incluyen la configuración de metadatos de objetos, la configuración de permisos y el cambio de la clase de almacenamiento de los objetos.

Puede utilizar también la operación de copia para copiar objetos existentes sin cifrar y escribirlos como objetos cifrados en el mismo bucket. Para obtener más información, consulte Cifrado de objetos con la herramienta de operaciones por lotes de Amazon S3.

Al copiar objetos, puede cambiar el algoritmo de suma de comprobación utilizado para calcular la suma de comprobación del objeto. Si los objetos no tienen calculada una suma de comprobación adicional, también puede agregarla especificando el algoritmo de suma de comprobación que utilizará Amazon S3. Para obtener más información, consulte Comprobación de la integridad de objetos .

Para obtener más información acerca de cómo copiar objetos en Amazon S3, así como los parámetros obligatorios y opcionales, consulte Copia, traslado y cambio de nombre de objetos en esta guía y CopyObject en la Referencia de la API de Amazon Simple Storage Service.

Restricciones y limitaciones

- Todos los objetos de origen deben estar en el mismo bucket.
- Todos los objetos de destino deben estar en el mismo bucket.
- Debe tener permisos de lectura en el bucket de origen y permisos de escritura en el bucket de destino.
- Los objetos que se van a copiar pueden tener un tamaño máximo de 5 GB.

- Si intenta copiar objetos de las clases S3 Glacier Flexible Retrieval o S3 Glacier Deep Archive a la clase de almacenamiento S3 Standard, primero debe restaurar estos objetos. Para obtener más información, consulte Restauración de un objeto archivado .
- Los trabajos de copia se deben crear en la región de destino, que es la región a la que pretende copiar los objetos.
- Se admiten todas las opciones de Copy, excepto para las comprobaciones condicionales de las ETag y el cifrado del lado del servidor con claves de cifrado proporcionadas por el cliente (SSE-C).
- Si los buckets no tienen control de versiones, se sobrescribirán los objetos con el mismo nombre de clave.
- Los objetos no se copian necesariamente en el mismo orden en el que aparecen en el manifiesto. Para los buckets con control de versiones, si es importante conservar el orden de versiones actual/no actual, primero debe copiar todas las versiones no actuales. Luego, una vez finalizado el primer trabajo, copie las versiones actuales en un trabajo posterior.
- No se admite la copia de objetos en la clase de almacenamiento de redundancia reducida (RRS).

Copia de objetos mediante operaciones por lotes de S3

Puede utilizar las operaciones por lotes de S3 para crear un trabajo de copia PUT para copiar objetos dentro de la misma cuenta o en otra cuenta de destino. Las siguientes secciones contienen ejemplos sobre cómo almacenar y utilizar un manifiesto que se encuentra en otra cuenta. En la primera sección, puede utilizar el inventario de Amazon S3 para entregar el informe de inventario a la cuenta de destino para utilizarlo durante la creación de trabajos o puede utilizar un manifiesto de valores separados por comas (CSV) en la cuenta de origen o de destino, como se muestra en el segundo ejemplo. En el tercer ejemplo, se muestra cómo utilizar la operación de copia para activar el cifrado de clave de bucket de S3 en objetos existentes.

Ejemplos de operaciones de copia

- Uso de un informe de inventario entregado a la cuenta de destino para copiar objetos entre Cuentas de AWS
- Uso de un manifiesto CSV para copiar objetos entre Cuentas de AWS
- Uso de la herramienta de operaciones por lotes de S3 para cifrar objetos con claves de bucket de S3

Uso de un informe de inventario entregado a la cuenta de destino para copiar objetos entre Cuentas de AWS

Utilice el inventario de Amazon S3 a fin de crear un informe de inventario y utilice este informe a fin de crear una lista de objetos para copiar con la herramienta de operaciones por lotes de S3. Para obtener más información sobre cómo utilizar un manifiesto CSV en la cuenta de origen o destino, consulte the section called “Uso de un manifiesto CSV para copiar objetos entre Cuentas de AWS”.

El inventario de Amazon S3 genera inventarios de los objetos de un bucket. La lista resultante se publica en un archivo saliente. El bucket cuyo inventario se crea se denomina bucket de origen y el bucket donde se almacena el archivo de informe de inventario se denomina bucket de destino.

Es posible configurar el informe de Amazon S3 Inventory para que se entregue a otra Cuenta de AWS. Esto permite a las operaciones por lotes de S3 leer el informe de inventario cuando se crea el trabajo en la cuenta de destino.

Para obtener más información sobre los buckets de origen y destino de inventario de Amazon S3, consulte Buckets de origen y destino.

La forma más sencilla de configurar un inventario es a través de la AWS Management Console, pero también puede utilizar la API REST, la AWS Command Line Interface (AWS CLI) o los SDK de AWS.

El siguiente procedimiento de la consola contiene los pasos de alto nivel para establecer permisos para un trabajo de Operaciones por lotes de S3. En este procedimiento se copian objetos de una cuenta de origen a una cuenta de destino, y el informe de inventario se almacena en la cuenta de destino.

Para configurar Amazon S3 Inventory para buckets de origen y destino pertenecientes a distintas cuentas

1. Inicie sesión en la AWS Management Console y abra la consola de Amazon S3 en https://console.aws.amazon.com/s3/.
2. Elija el bucket de destino en el que desea almacenar el informe de inventario.

   Elija un bucket de manifiestos de destino para almacenar el informe de inventario. En este procedimiento, la cuenta de destino es la cuenta a la que pertenecen tanto el bucket de manifiestos de destino como el bucket en el que se copian los objetos.
3. Configure un inventario para enumerar los objetos de un bucket de origen y publicar la lista en el bucket de manifiestos de destino.

Configure una lista de inventario para un bucket de origen. Cuando lo haga, especifique el bucket de destino donde desea que se almacene la lista. El informe de inventario para el bucket de origen se publica en el bucket de destino. En este procedimiento, la cuenta de origen es la cuenta propietaria del bucket de origen.

Para obtener información acerca de cómo utilizar la consola para configurar un inventario o cómo cifrar un archivo de lista de inventario, consulte Configuración de Inventario de Amazon S3.

Elija CSV para el formato de salida.

Cuando introduzca la información del bucket de destino, elija Buckets in another account (Los buckets de otra cuenta). A continuación, introduzca el nombre del bucket de manifiestos de destino. Si lo desea, puede introducir el ID de la cuenta de destino.

Cuando se guarda la configuración de inventario, la consola muestra un mensaje similar al siguiente:

Amazon S3 could not create a bucket policy on the destination bucket. Ask the destination bucket owner to add the following bucket policy to allow Amazon S3 to place data in that bucket (Amazon S3 no pudo crear una política de bucket en el bucket de destino. Pida al propietario del bucket de destino que añada la siguiente política de bucket para permitir que Amazon S3 coloque datos en ese bucket.

A continuación, la consola muestra una política de bucket que se puede utilizar para el bucket de destino.

4. Copie la política de bucket de destino que aparece en la consola.
5. En la cuenta de destino, añada la política de bucket que ha copiado al bucket de manifiestos de destino donde se almacena el informe de inventario.
6. En la cuenta de destino, cree un rol basado en la política de confianza de Operaciones por lotes de S3. Para obtener más información acerca de la política de confianza, consulte Política de confianza.

   Para obtener más información acerca de cómo crear un rol, consulte Creación de un rol para delegar permisos a un servicio de AWS en la Guía del usuario de IAM.

Introduzca un nombre para el rol (el rol de ejemplo utiliza el nombre `BatchOperationsDestinationRoleCOPY`). Elija el servicio S3 y, a continuación, elija el caso de uso S3 bucket Batch Operations (Operaciones por lotes de bucket de S3), que aplica la política de confianza al rol.

A continuación, elija Create policy (Crear política) para asociar la política siguiente al rol.

```
{
  "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/*"
      ]
    }
  ]
}
```

El rol utiliza la política para conceder permiso a `batchoperations.s3.amazonaws.com` para leer el manifiesto en el bucket de destino. También concede permisos GET para objetos, listas de control de acceso (ACL), etiquetas y versiones en el bucket de objetos de origen. Y concede permisos PUT para objetos, ACL, etiquetas y versiones en el bucket de objetos de destino.

7. En la cuenta de origen, cree una política de bucket para el bucket de origen que otorgue el rol que creó en el paso anterior para obtener (GET) objetos, ACL, etiquetas y versiones en el bucket de origen. Este paso permite a Operaciones por lotes de S3 obtener objetos del bucket de origen a través del rol de confianza.

   A continuación, se muestra un ejemplo de política de bucket para la cuenta de origen.

```
{
  "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. Una vez que el informe de inventario esté disponible, cree un trabajo de copia del objeto PUT de Operaciones por lotes de S3 en la cuenta de destino y seleccione el informe de inventario en el bucket de manifiestos de destino. Necesita el ARN del rol que creó en la cuenta de destino.

   Para obtener información general acerca de cómo crear un trabajo, consulte Creación de trabajos de operaciones por lotes de S3.

   Para obtener información acerca de cómo crear un trabajo mediante la consola, consulte Creación de trabajos de operaciones por lotes de S3.

Uso de un manifiesto CSV para copiar objetos entre Cuentas de AWS

Puede utilizar un manifiesto CSV almacenado en la cuenta de origen para copiar objetos entre Cuentas de AWS con Operaciones por lotes de Amazon S3. Para usar un informe de inventario de S3 como manifiesto, consulte the section called “Uso de un informe de inventario para copiar objetos entre Cuentas de AWS”.

Para ver un ejemplo del formato CSV de los archivos de manifiesto, consulte the section called “Creación de un archivo de manifiesto”.

En el siguiente procedimiento, se muestra cómo configurar los permisos cuando se utiliza un trabajo de Operaciones por lotes de Amazon S3 para copiar objetos desde una cuenta de origen a una cuenta de destino con el archivo de manifiesto CSV almacenado en la cuenta de origen.

Uso de un manifiesto CSV para copiar objetos entre Cuentas de AWS

1. En la cuenta de destino, cree un rol basado en la política de confianza de Operaciones por lotes de S3. En este procedimiento, la cuenta de destino es la cuenta en la que se copian los objetos.

   Para obtener más información acerca de la política de confianza, consulte Política de confianza.

   Para obtener más información acerca de cómo crear un rol, consulte Creación de un rol para delegar permisos a un servicio de AWS en la Guía del usuario de IAM.

   Si crea el rol utilizando la consola, introduzca un nombre para el rol (el rol de ejemplo utiliza el nombre `BatchOperationsDestinationRoleCOPY`). Elija el servicio S3 y, a continuación, elija el caso de uso S3 bucket Batch Operations (Operaciones por lotes de bucket de S3), que aplica la política de confianza al rol.

   A continuación, elija Crear política para asociar la política siguiente al rol:

```
{
  "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/*"
      ]
    }
  ]
}
```

Mediante la política, el rol concede permiso a `batchoperations.s3.amazonaws.com` para leer el manifiesto en el bucket de manifiestos de origen. Concede permisos para objetos `GET`, listas de control de acceso (ACL), etiquetas y versiones en el bucket de objetos de origen. También concede permisos para objetos `PUT`, ACL, etiquetas y versiones en el bucket de objetos de destino.

2. En la cuenta de origen, cree una política de bucket para el bucket que contenga el manifiesto para conceder el rol que creó en el paso anterior para objetos `GET` y versiones en el bucket de manifiesto de origen.

   Este paso permite a Operaciones por lotes de Amazon S3 leer el manifiesto utilizando el rol de confianza. Aplique la política de bucket al bucket que contiene el manifiesto.

   A continuación, se muestra un ejemplo de la política de bucket que puede aplicarse al bucket de manifiesto de origen.

```
{
  "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/*"
    }
  ]
}
```

Esta política también concede permisos para permitir a un usuario de la consola que esté creando un trabajo en la cuenta de destino los mismos permisos en el bucket de manifiestos de origen a través de la misma política de bucket.

3. En la cuenta de origen, cree una política de bucket para el bucket de origen que conceda al rol que creó permisos para objetos GET, ACL, etiquetas y versiones en el bucket de objetos de origen. Operaciones por lotes de S3 puede, entonces, obtener objetos del bucket de origen a través del rol de confianza.

   A continuación, se muestra un ejemplo de política de bucket para el bucket que contiene los objetos de origen:

```
{
  "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. Cree un trabajo de Operaciones por lotes de S3 en la cuenta de destino. Necesita el nombre de recurso de Amazon (ARN) para el rol que ha creado en la cuenta de destino. Para obtener más información acerca de la creación de un trabajo, consulte Creación de trabajos de operaciones por lotes de S3.

Uso de la herramienta de operaciones por lotes de S3 para cifrar objetos con claves de bucket de S3

En esta sección, se utiliza la operación de copia de la herramienta de operaciones por lotes de Amazon S3 para identificar y activar el cifrado de claves de bucket de S3 en objetos existentes. Para obtener más información sobre las claves de bucket de S3, consulte Reducción del costo de SSE-KMS con las claves de bucket de Amazon S3 y Configuración del bucket para utilizar una clave de bucket de S3 con SSE-KMS para objetos nuevos.

Entre los temas que se tratan en este ejemplo, se incluyen los siguientes:

Temas

- Requisitos previos
- Paso 1: obtener la lista de objetos mediante Amazon S3 Inventory
- Paso 2: filtrar la lista de objetos con S3 Select
- Paso 3: configurar y ejecutar el trabajo de la herramienta de operaciones por lotes de S3
- Resumen

Requisitos previos

Para seguir los pasos de este procedimiento, necesita una Cuenta de AWS y, al menos, un bucket de S3 para almacenar los archivos de trabajo y los resultados cifrados. También le puede resultar útil gran parte de la documentación existente de la herramienta de operaciones por lotes de S3, incluidos los siguientes temas:

- Conceptos básicos de Operaciones por lotes de S3
- Creación de trabajos de operaciones por lotes de S3

- Operaciones compatibles con las operaciones por lotes de S3
- Administración de trabajos de operaciones por lotes de S3

Paso 1: obtener la lista de objetos mediante Amazon S3 Inventory

Para empezar, identifique el bucket de S3 que contiene los objetos que desea cifrar y obtenga una lista del contenido. Un informe de inventario de Amazon S3 es la forma más conveniente y asequible de hacerlo. En el informe se proporciona la lista de los objetos de un bucket junto con los metadatos asociados. El bucket de origen hace referencia al bucket cuyo inventario se crea y el bucket de destino hace referencia al bucket donde se almacena el archivo de informe de inventario. Para obtener más información sobre los buckets de origen y destino de inventario de Amazon S3, consulte Inventario de Amazon S3.

La forma más sencilla de configurar un inventario es usar la AWS Management Console. Sin embargo, puede también usar la API REST, la AWS Command Line Interface (AWS CLI) o los SDK de AWS. Antes de seguir estos pasos, inicie sesión en la consola y abra la consola de Amazon S3 en https://console.aws.amazon.com/s3/. Si encuentra errores de permiso denegado, agregue una política de bucket al bucket de destino. Para obtener más información, consulte Concesión de permisos para el inventario de S3 y el análisis de S3.

Para obtener la lista de objetos mediante el inventario de S3

1. Abra la consola de Amazon S3 en https://console.aws.amazon.com/s3.
2. En el panel de navegación, elija Buckets y, luego, elija un bucket que contenga objetos para cifrar.
3. En la pestaña Management (Administración), diríjase a la sección Inventory configurations (Configuraciones de inventario) y elija Create inventory configuration (Crear una configuración de inventario).
4. Asigne un nombre a su nuevo inventario, introduzca el nombre del bucket de S3 de destino y, opcionalmente, cree un prefijo de destino para que Amazon S3 asigne objetos en ese bucket.
5. Elija CSV para Output format (Formato de salida).
6. (Opcional) En la sección Campos adicionales: opcional, elija Cifrado y cualquier otro campo de informe que le interese. Establezca la frecuencia de entregas de informes en Daily (Diaria) para que el primer informe se entregue a su bucket más pronto.
7. Elija Save (Guardar) para guardar la configuración.

Amazon S3 puede tardar hasta 48 horas en entregar el primer informe, por lo que debe verificar cuando este llegue. Después de recibir el primer informe, diríjase a la siguiente sección para filtrar el contenido del informe de Inventario de S3. Si ya no quiere recibir informes de inventario para este bucket, elimine la configuración de inventario de S3. De lo contrario, S3 entrega informes de manera diaria o semanal.

Una lista de inventario no es una vista estática de todos los objetos. Las listas de inventario son una instantánea continua de los elementos de un bucket, que son coherentes en última instancia (por ejemplo: la lista podría no incluir los objetos agregados o eliminados más recientemente). La combinación de la herramienta de operaciones por lotes de S3 y del inventario de S3 funciona mejor cuando se trabaja con objetos estáticos o con un conjunto de objetos creado hace dos o más días. Con el fin de trabajar con datos más recientes, utilice la operación de la API ListObjectsv2 (GET Bucket) para crear la lista de objetos de forma manual. Si es necesario, repita el proceso durante los próximos días o hasta que el informe de inventario muestre el estado deseado para todas las claves.

Paso 2: filtrar la lista de objetos con S3 Select

Después de recibir el informe de Inventario de S3, puede filtrar su contenido para mostrar solo los objetos que no estén cifrados con Claves de bucket de S3. Si desea que todos los objetos del bucket se cifren con Claves de bucket de S3, puede omitir este paso. Sin embargo, al filtrar el informe de inventario de S3 en esta fase, se ahorra tiempo y gastos de volver a cifrar objetos que ya había cifrado.

Aunque en los siguientes pasos se muestra cómo filtrar mediante Amazon S3 Select, también puede utilizar Amazon Athena. Para decidir qué herramienta usar, consulte el archivo `manifest.json` del informe de inventario de S3. En este archivo se muestra el número de archivos de datos asociados a ese informe. Si el número es grande, utilice Amazon Athena, ya que se ejecuta en varios objetos de S3, mientras que S3 Select funciona solo en un objeto a la vez. Para obtener más información sobre cómo usar Amazon S3 y Athena juntos, consulte.Consulta de Amazon S3 Inventory con Amazon Athena y Using Athena en la publicación de blog Encrypting objects with Amazon S3 Batch Operations.

Para filtrar el informe de inventario de S3 mediante S3 Select

1. Abra el archivo `manifest.json` del informe de inventario y consulte la sección `fileSchema` del JSON. Esta es información para la consulta que se ejecuta en los datos.

El siguiente JSON es un ejemplo del archivo `manifest.json` para un inventario con formato CSV en un bucket con el control de versiones habilitado. Dependiendo de cómo haya configurado el informe de inventario, el manifiesto podría tener un aspecto diferente.

```
  {
    "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 el control de versiones no está activado en el bucket, o si decide ejecutar el informe para las versiones más recientes, el `fileSchema` es `Bucket`, `Key`, y `BucketKeyStatus`.

Si el control de versiones está activado, dependiendo de cómo configure el informe de inventario, el `fileSchema` podría incluir lo siguiente: `Bucket`, `Key`, `VersionId`, `IsLatest`, `IsDeleteMarker`, `BucketKeyStatus`. Por lo tanto, preste atención a las columnas 1, 2, 3 y 6 cuando ejecute su consulta.

Operaciones por lotes de S3 necesita el bucket, la clave y el ID de versión como entradas para realizar el trabajo, además del campo por el que se debe buscar, que es `BucketKeyStatus`. No necesita el campo ID de versión, pero especificarlo es útil cuando opera en un bucket con control de versiones. Para obtener más información, consulte Trabajar con objetos en un bucket con control de versiones habilitado.

2. Ubique los archivos de datos para el informe de inventario. El objeto `manifest.json` muestra los archivos de datos en files (archivos).
3. Después de ubicar y seleccionar el archivo de datos en la consola de S3, elija Actions (Acciones) y, luego, Query with S3 Select (Consultar con S3 Select).

4. Mantenga los campos preseleccionados CSV, Comma (Coma), y GZIP dentro de la selección y elija Next (Siguiente).
5. Para revisar el formato del informe de inventario antes de continuar, seleccione Show file preview (Mostrar previsualización del archivo).
6. Introduzca las columnas a las que se debe hacer referencia en el campo de expresión SQL. Luego, elija Run SQL (Ejecutar SQL). La siguiente expresión muestra las columnas 1 a 3 para todos los objetos sin una clave de bucket de S3 configurada.

   ```
   select s._1, s._2, s._3 from s3object s where s._6 = 'DISABLED'
   ```

   A continuación se incluyen resultados de ejemplo.

   ```
         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. Descargue los resultados, guárdelos en un formato CSV y cárguelos en Amazon S3 como la lista de objetos para el trabajo de la herramienta de operaciones por lotes de S3.
8. Si tiene varios archivos de manifiesto, ejecute Query with S3 Select (Consultar con S3 Select) en esos también. Dependiendo del tamaño de los resultados, puede combinar las listas y ejecutar un único trabajo de la herramienta de operaciones por lotes de S3 o ejecutar cada lista como un trabajo independiente.

   Considere el precio de ejecutar cada trabajo de la herramienta de operaciones por lotes de S3 cuando decida el número de trabajos que se van a ejecutar.

Paso 3: configurar y ejecutar el trabajo de la herramienta de operaciones por lotes de S3

Ahora que tiene sus listas CSV filtradas de objetos de S3, puede comenzar el trabajo de la herramienta de operaciones por lotes de S3 para cifrar los objetos con claves de bucket de S3.

Un trabajo hace referencia colectivamente a la lista (manifiesto) de objetos proporcionados, la operación realizada y los parámetros especificados. La forma más sencilla de cifrar este conjunto de objetos es usar la operación de copia PUT y especificar el mismo prefijo de destino que los objetos que aparecen en el manifiesto. Esto sobrescribe los objetos existentes en un bucket sin control de versiones o, con el control de versiones activado, crea una versión más reciente y cifrada de los objetos.

Como parte de la copia de los objetos, especifique que Amazon S3 debe cifrar el objeto con cifrado SSE-KMS y S3. Este trabajo copia los objetos, por lo que todos los objetos muestran una fecha de creación actualizada al finalizar, independientemente de cuándo los agregó originalmente a S3. Especifique también las otras propiedades del conjunto de objetos como parte del trabajo de la herramienta de operaciones por lotes de S3, incluidas las etiquetas de objeto y la clase de almacenamiento.

Pasos secundarios

- Configuración de la política de IAM
- Configuración del rol de IAM de la herramienta de operaciones por lotes
- Activación de las claves de bucket de S3 para un bucket existente
- Creación de un trabajo de la herramienta de operaciones por lotes
- Ejecución del trabajo de la herramienta de operaciones por lotes
- Puntos a tener en cuenta

Configuración de la política de IAM

1. Abra la consola de IAM en https://console.aws.amazon.com/iam/.
2. En el panel de navegación, seleccione Policies (Políticas) y, a continuación, seleccione Create Policy (Crear política).
3. Seleccione la pestaña JSON. Seleccione Edit policy (Editar política) y agregue la política de IAM de ejemplo que aparece en el siguiente bloque de código.

   Después de copiar el ejemplo de política en la consola de IAM, reemplace lo siguiente:

   a. Reemplace *SOURCE_BUCKET_FOR_COPY* con el nombre del bucket de origen.
   b. Reemplace *DESTINATION_BUCKET_FOR_COPY* con el nombre del bucket de destino.
   c. Reemplace *MANIFEST_KEY* con el nombre del objeto de manifiesto.
   d. Reemplace *REPORT_BUCKET* con el nombre del bucket donde desee guardar los informes.

```
{
  "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. Elija Next: Tags (Siguiente: etiquetas).
5. Agregue las etiquetas que desee (opcional) y elija Next: Review (Siguiente: revisar).
6. Proporcione un nombre para la política y, opcionalmente, una descripción; a continuación, elija Create policy (Crear política).
7. Elija Review policy (Revisar política) y, a continuación, Save changes (Guardar cambios).
8. Con su política de la herramienta de operaciones por lotes de S3 ya completada, la consola lo lleva nuevamente a la página de IAM Policies (Políticas). Filtre por el nombre de la política, elija el botón situado a la izquierda del nombre de la política, elija Policy actions (Acciones de política) y, a continuación, Attach (Adjuntar).

   Para adjuntar la política recién creada a un rol de IAM, seleccione los usuarios, grupos o roles apropiados en la cuenta y elija Attach policy (Adjuntar política). Esto lo lleva de vuelta a la consola de IAM.

Configuración del rol de IAM de la herramienta de operaciones por lotes

1. En el panel de navegación de la consola de IAM, seleccione Roles y, a continuación, seleccione Crear rol.
2. Elija Servicio de AWS, S3 y Operaciones por lotes de S3. A continuación, elija Next: Permissions (Siguiente: permisos).
3. Comience a escribir el nombre de la policy (política) del IAM que acaba de crear. Seleccione la casilla de verificación junto al nombre de la política cuando aparezca y elija Next: Tags (Siguiente: etiquetas).
4. (Opcional) Agregue etiquetas o mantenga los campos de clave y valor en blanco para este ejercicio. Elija Next: Review (Siguiente: revisar).
5. Introduzca un nombre de rol y acepte la descripción predeterminada o agregue la suya. Elija Create role (Crear rol).
6. Asegúrese de que el usuario que crea el trabajo tenga los permisos del siguiente ejemplo.

   Reemplace *{ACCOUNT-ID}* con su ID de Cuenta de AWS y *{IAM_ROLE_NAME}* con el nombre que tiene previsto aplicar al rol de IAM que creará más adelante en el paso de creación de trabajo de Operaciones por lotes. Para obtener más información, consulte Concesión de permisos para Operaciones por lotes de S3.

```
            {
            "Sid": "AddIamPermissions",
            "Effect": "Allow",
            "Action": [
            "iam:GetRole",
            "iam:PassRole"
            ],
            "Resource": "arn:aws:iam::ACCOUNT-ID:role/IAM_ROLE_NAME"
            }
```

Activación de las claves de bucket de S3 para un bucket existente

1. Abra la consola de Amazon S3 en https://console.aws.amazon.com/s3.
2. En la lista Buckets, elija el bucket para el que desea habilitar una clave de bucket de S3.
3. Seleccione Properties (Propiedades).
4. En Default encryption (Cifrado predeterminado), elija Edit (Editar).
5. En Tipo de cifrado, elija Claves administradas de Amazon S3 (SSE-S3) o Clave de AWS Key Management Service (SSE-KMS).
6. Si elige la Clave de AWS Key Management Service (SSE-KMS), en AWS KMS key puede especificar su clave de AWS KMS mediante una de las siguientes opciones.
   - Para elegir de una lista de claves de KMS disponibles, marque Elija entre sus claves de AWS KMS. En la lista de claves disponibles, elija una clave de KMS de cifrado simétrica de la misma región que el bucket. En esta lista aparecen tanto la clave administrada de AWS (`aws/s3`) como las claves administradas por el cliente.
   - Para introducir el ARN de la clave de KMS, elija Introducir el ARN de la clave de AWS KMS e introduzca el ARN de la clave de KMS en el campo que aparece.
   - Para crear una nueva clave administrada por el cliente en la consola de AWS KMS, elija Crear una clave de KMS.
7. En Bucket key (Clave de bucket), elija Enable (Habilitar) y, luego, Save changes (Guardar cambios).

Ahora que la clave de bucket de S3 está activada en el nivel de bucket, los objetos que se carguen, modifiquen o copien en este bucket heredarán esta configuración de cifrado de forma

predeterminada. Aquí también se incluyen los objetos copiados con Operaciones por lotes de Amazon S3.

Creación de un trabajo de la herramienta de operaciones por lotes

1. Abra la consola de Amazon S3 en https://console.aws.amazon.com/s3.
2. En el panel de navegación, elija Batch Operations (Operaciones por lotes) y, luego, Create Job (Crear trabajo).
3. Elija la Region (Región) donde almacena los objetos y, a continuación, CSV como el tipo de manifiesto.
4. Introduzca la ruta o desplácese hasta el archivo de manifiesto CSV que creó antes a partir de los resultados de S3 Select (o Athena). Si el manifiesto contiene ID de versiones, seleccione esa casilla de verificación. Elija Next (Siguiente).
5. Elija la operación Copy (Copiar) y elija el bucket de destino de copia. Puede mantener el cifrado del lado del servidor desactivado. Siempre y cuando el bucket de destino tenga habilitadas las claves de bucket de S3, la operación de copia aplica claves de bucket de S3 en el bucket de destino.
6. (Opcional) Elija una clase de almacenamiento y los demás parámetros que desee. Los parámetros que especifique en este paso se aplican a todas las operaciones realizadas en los objetos que aparecen en el manifiesto. Seleccione Siguiente.
7. Para configurar el cifrado del servidor, siga los siguientes pasos:

   a. En Cifrado del lado del servidor, elija una de las siguientes opciones:

      - Para mantener la configuración del bucket para el cifrado predeterminado del servidor de los objetos al almacenarlos en Amazon S3, elija No especifique una clave de cifrado. Siempre y cuando el bucket de destino tenga habilitadas las claves de bucket de S3, la operación de copia aplicará una clave de bucket de S3 al bucket de destino.

        > **Note**
        >
        > Si la política de bucket para el destino especificado exige que los objetos estén cifrados antes de almacenarlos en Amazon S3, debe especificar una clave de cifrado. De lo contrario, se producirá un error al copiar los objetos en el destino.

      - Para cifrar objetos antes de almacenarlos en Amazon S3, elija Especificar una clave de cifrado.

b. En Configuración del cifrado, si selecciona Especificar una clave de cifrado, debe elegir entre Usar la configuración del bucket de destino para el cifrado predeterminado o Anular la configuración del bucket de destino para el cifrado predeterminado.

c. Si elige Anular la configuración del bucket de destino para el cifrado predeterminado, debe configurar los siguientes ajustes de cifrado.

   i. En Tipo de cifrado, elija Claves administradas de Amazon S3 (SSE-S3) o Clave de AWS Key Management Service (SSE-KMS). SSE-S3 utiliza uno de los cifrados de bloques más seguros, Advanced Encryption Standard de 256 bits (AES-256), para cifrar cada objeto. SSE-KMS le proporciona más control sobre su clave. Para obtener más información, consulte Uso del cifrado del servidor con claves administradas por Amazon S3 (SSE-S3) y Uso del cifrado del servidor con claves de AWS KMS (SSE-KMS).

   ii. Si elige Clave de AWS Key Management Service (SSE-KMS), en AWS KMS key puede especificar su AWS KMS key mediante una de las siguientes opciones.

   - Para elegir de entre una lista de claves de KMS disponibles, marque Elija entre sus claves de AWS KMS keys y seleccione una clave de KMS de cifrado simétrica de la misma región que el bucket. En esta lista aparecen tanto la clave administrada de AWS (`aws/s3`) como las claves administradas por el cliente.
   - Para introducir el ARN de la clave de KMS, elija Introducir el ARN de la clave de AWS KMS e introduzca el ARN de la clave de KMS en el campo que aparece.
   - Para crear una nueva clave administrada por el cliente en la consola de AWS KMS, elija Crear una clave de KMS.

   iii. En Bucket Key (Clave de bucket), seleccione Enable (Habilitar). La operación de copia aplica una clave de bucket de S3 al bucket de destino.

8. Proporcione a su trabajo una descripción (o mantenga el valor predeterminado), establezca su nivel de prioridad, elija un tipo de informe y especifique la ruta al destino del informe de finalización.

9. En la sección Permissions (Permisos), asegúrese de elegir el rol de IAM de la herramienta de operaciones por lotes que definió antes. Elija Next (Siguiente).

10. En Review (Revisar), compruebe la configuración. Si desea realizar cambios, seleccione Previous (Anterior). Después de confirmar la configuración de la herramienta de operaciones por lotes, elija Create job (Crear trabajo).

    Para obtener más información, consulte Creación de trabajos de operaciones por lotes de S3.

Ejecución del trabajo de la herramienta de operaciones por lotes

El asistente de configuración lo redirecciona automáticamente a la sección Operaciones por lotes de S3 de la consola de Amazon S3. El trabajo nuevo pasa del estado New (Nuevo) al estado Preparing (En preparación) cuando S3 comienza el proceso. Cuando el estado es “Preparing” (En preparación), S3 lee el manifiesto del trabajo, comprueba si hay errores y calcula el número de objetos.

1. Elija el botón de actualización en la consola de Amazon S3 para comprobar el progreso. Dependiendo del tamaño del manifiesto, la lectura puede tardar minutos u horas.
2. Después de que S3 termina de leer el manifiesto del trabajo, este pasa al estado Awaiting your confirmation (A la espera de confirmación). Elija el botón de opción a la izquierda del ID de trabajo y elija Run job (Ejecutar trabajo).
3. Verifique la configuración del trabajo y elija Run job (Ejecutar trabajo) en la esquina inferior derecha.

   Una vez que el trabajo comience a ejecutarse, puede seleccionar el botón de actualización para verificar el progreso a través de la vista del panel de consola o mediante la selección de un trabajo específico.
4. Una vez completado el trabajo, podrá ver el recuento de objetos Successful (Correctos) y Failed (Con errores) para confirmar que todo se realizó como se esperaba. Si ha habilitado los informes de trabajos, verifique en estos la causa exacta de las operaciones con errores.

   También puede llevar a cabo estos pasos mediante la AWS CLI, los SDK de AWS o la API de REST de Amazon S3. Para obtener más información acerca del seguimiento del estado del trabajo y los informes de finalización, consulte Seguimiento del estado del trabajo e informes de finalización.

Puntos a tener en cuenta

Tenga en cuenta los siguientes problemas cuando utilice Operaciones por lotes de S3 para cifrar objetos con claves de bucket de S3:

- Se le cobrarán los trabajos, los objetos y las solicitudes de Operaciones por lotes de S3, además de los cargos asociados a la operación que Operaciones por lotes de S3 realiza en su nombre, incluidas las transferencias de datos, las solicitudes y otros cargos. Para obtener más información, consulte Precios de Amazon S3.
- Si utiliza un bucket con control de versiones, cada trabajo de la herramienta de operaciones por lotes de S3 realizado crea nuevas versiones cifradas de los objetos. También mantiene las

versiones anteriores sin una clave de bucket de S3 configurada. Para eliminar las versiones anteriores, configure una política de vencimiento del ciclo de vida de S3 para las versiones no actuales como se describe en Elementos de configuración del ciclo de vida.
- La operación de copia crea nuevos objetos con nuevas fechas de creación, lo que puede afectar a las acciones del ciclo de vida, como el archivado. Si copia todos los objetos del bucket, todas las copias nuevas tendrán fechas de creación idénticas o similares. Para identificar aún más estos objetos y crear reglas de ciclo de vida diferentes para varios subconjuntos de datos, considere la posibilidad de utilizar etiquetas de objeto.

Resumen

En esta sección, ordenó los objetos existentes para filtrar los datos ya cifrados. A continuación, aplicó la característica de clave de bucket de S3 a objetos no cifrados con Operaciones por lotes de S3 para copiar los datos existentes en un bucket con la clave de bucket de S3 activada. Este proceso le puede ahorrar tiempo y dinero a la vez que le permite completar operaciones como el cifrado de todos los objetos existentes.

Para obtener más información sobre la herramienta de operaciones por lotes de S3, consulte Realización de operaciones por lotes a gran escala en objetos de Amazon S3.

Para obtener ejemplos en los que se muestra la operación de copia con etiquetas mediante la AWS CLI y AWS SDK for Java, consulte Creación de un trabajo de Operaciones por lotes con etiquetas de trabajo como etiquetado.

Invocar a la función AWS Lambda

La invocación de la función de AWS Lambda inicia las funciones de AWS Lambda para que realicen acciones personalizadas en objetos que aparecen en un manifiesto. En esta sección, se describe cómo crear una función Lambda para usarla con Operaciones por lotes de S3 y cómo crear un trabajo para invocar la función. El trabajo de Operaciones por lotes de S3 utiliza la operación `LambdaInvoke` para ejecutar una función Lambda en cada objeto que aparece en un manifiesto.

Puede trabajar con la herramienta de operaciones por lotes de S3 para Lambda mediante la AWS Management Console, AWS Command Line Interface (AWS CLI), los SDK de AWS o las API REST. Para obtener más información acerca del uso de Lambda, consulte Introducción a AWS Lambda en la Guía para desarrolladores de AWS Lambda.

En las siguientes secciones, se explica cómo puede comenzar a usar Operaciones por lotes de S3 con Lambda.

Temas

Uso de Lambda con Operaciones por lotes de Amazon S3

Cuando utilice la herramienta de operaciones por lotes de S3 con AWS Lambda, debe crear nuevas funciones de Lambda específicamente para utilizarlas con la herramienta de operaciones por lotes de S3. No puede reutilizar funciones basadas en eventos de Amazon S3 existentes con Operaciones por lotes de S3. Las funciones de eventos solo pueden recibir mensajes; no devuelven mensajes. Las funciones Lambda que se utilizan con Operaciones por lotes de S3 deben aceptar y devolver mensajes. Para obtener más información acerca del uso de Lambda con los eventos de Amazon S3, consulte Uso de AWS Lambda con Amazon S3 en la Guía para desarrolladores de AWS Lambda.

Debe crear un trabajo de Operaciones por lotes de S3 que invoque a su función Lambda. El trabajo ejecuta la misma función Lambda en todos los objetos que aparecen en el manifiesto. Puede controlar qué versiones de su función Lambda se deben usar mientras procesa los objetos de su manifiesto. Operaciones por lotes de S3 admite nombres de recursos de Amazon (ARN) no calificados, alias y versiones específicas. Para obtener más información, consulte Introducción al control de versiones de AWS Lambda en la Guía para desarrolladores de AWS Lambda.

Si proporciona el trabajo de Operaciones por lotes de S3 con un ARN de función que utiliza un alias o el calificador `$LATEST`, y actualiza la versión a la que apunta cualquiera de ellos, Operaciones por lotes de S3 comenzará a llamar a la nueva versión de su función Lambda. Esto puede resultar útil cuando se desea actualizar la funcionalidad en medio de un trabajo grande. Si no quiere que Operaciones por lotes de S3 cambie la versión que se utiliza, facilite la versión específica en el parámetro `FunctionARN` al crear el trabajo.

Uso de Lambda y Operaciones por lotes de Amazon S3 con buckets de directorio

Los buckets de directorio son un tipo de bucket de Amazon S3 que está diseñado para cargas de trabajo o aplicaciones de rendimiento crítico que requieren una latencia constante de milisegundos de un solo dígito. Para obtener más información, consulte Buckets de directorio.

Existen requisitos especiales para utilizar las operaciones por lotes de Amazon S3 para invocar funciones de Lambda que actúan en los buckets de directorio. Por ejemplo, debe estructurar la solicitud de Lambda mediante un esquema JSON actualizado y especificar `InvocationSchemaVersion` **`2.0`** cuando se crea el trabajo. Este esquema actualizado le permite especificar pares clave-valor opcionales para `UserArguments`, lo que puede modificar determinados parámetros de las funciones de Lambda existentes. Para obtener más información, consulte Automate object processing in Amazon S3 directory buckets with S3 Batch Operations and AWS Lambda en el AWS Storage Blog.

Códigos de respuesta y de resultados

Operaciones por lotes de S3 invoca la función de Lambda con una o más claves, cada una de las cuales tiene un `TaskID` asociado. Operaciones por lotes de S3 espera un código de resultados por clave de las funciones de Lambda. Cualquier ID de tarea enviado en la solicitud que no se devuelva con un código de resultado por clave recibirá el código de resultado del campo `treatMissingKeysAs`. `treatMissingKeysAs` es un campo de solicitud opcional y su valor predeterminado es `TemporaryFailure`. La siguiente tabla contiene los demás códigos de resultado y valores posibles para el campo `treatMissingKeysAs`.

Código de respuesta	Descripción
`Succeeded`	La tarea se completó normalmente. Si solicitó un informe de finalización de trabajos, la cadena de resultados de la tarea se incluye en el informe.
`TemporaryFailure`	Se detectó un error temporal en la tarea y esta se redirigirá antes de que se complete el trabajo. La cadena de resultados se pasa por alto. Si este es el último redireccionamiento, el mensaje de error se incluye en el informe final.
`PermanentFailure`	Se detectó un error permanente en la tarea. Si solicitó un informe de finalización de trabajos, la tarea se marca como `Failed` e incluye la cadena del mensaje de error. Las cadenas de resultados de tareas con error se pasan por alto.

Creación de una función Lambda para utilizarla con Operaciones por lotes de S3

En esta sección, se proporcionan ejemplos de permisos de AWS Identity and Access Management (IAM) que debe utilizar con su función de Lambda. También contiene una función Lambda de ejemplo que se puede utilizar con Operaciones por lotes de S3. Si nunca ha creado una función de Lambda, consulte el Tutorial: uso de AWS Lambda con Amazon S3 en la Guía para desarrolladores de AWS Lambda.

Debe crear funciones Lambda para utilizarlas específicamente con Operaciones por lotes de S3. No puede reutilizar las funciones Lambda basadas en eventos de Amazon S3 existentes. Esto se debe a que las funciones Lambda que se utilizan para Operaciones por lotes de S3 deben aceptar y devolver campos de datos especiales.

> **Important**
>
> Las funciones de AWS Lambda escritas en Java aceptan las interfaces de controlador RequestHandler o RequestStreamHandler. Sin embargo, con el fin de admitir el formato de solicitud y respuesta de la herramienta de operaciones por lotes de S3, AWS Lambda necesita la interfaz `RequestStreamHandler` para la serialización y la deserialización personalizadas de una solicitud y su respuesta. Esta interfaz permite a Lambda pasar un InputStream y OutputStream al método `handleRequest` de Java.
> Asegúrese de usar la interfaz `RequestStreamHandler` cuando utilice funciones Lambda con Operaciones por lotes de S3. Si utiliza una interfaz `RequestHandler`, el trabajo por lotes producirá un error y, en el informe de finalización, aparecerá un mensaje que indica que se ha devuelto un JSON no válido en la carga útil de Lambda.
> Para obtener más información, consulte Interfaces de controlador en la Guía del usuario de AWS Lambda.

Permisos de IAM de ejemplo

A continuación, se muestran ejemplos de los permisos de IAM necesarios para utilizar una función Lambda con Operaciones por lotes de S3.

Example - Política de confianza de Operaciones por lotes de S3

Este es un ejemplo de la política de confianza que puede usar para el rol de IAM de Operaciones por lotes. Este rol de IAM se especifica al crear el trabajo y ofrece a Operaciones por lotes permiso para asumirlo.

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

Example - Política de IAM de Lambda

A continuación, se muestra un ejemplo de una política de IAM que da permiso a Operaciones por lotes de S3 para invocar a la función Lambda y leer el manifiesto de entrada.

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Sid": "BatchOperationsLambdaPolicy",
            "Effect": "Allow",
            "Action": [
                "s3:GetObject",
                "s3:GetObjectVersion",
                "s3:PutObject",
                "lambda:InvokeFunction"
            ],
            "Resource": "*"
        }
    ]
}
```

Solicitud y respuesta de ejemplo

Esta sección contiene ejemplos de solicitudes y respuestas para la función Lambda.

Example Solicitud

A continuación, se muestra un ejemplo de JSON de una solicitud para la función 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 Respuesta

A continuación, se muestra un ejemplo de JSON de una respuesta para la función Lambda.

```
{
  "invocationSchemaVersion": "1.0",
  "treatMissingKeysAs" : "PermanentFailure",
  "invocationId" : "YXNkbGZqYWRmaiBhc2RmdW9hZHNmZGpmaGFzbGtkaGZza2RmaAo",
  "results": [
    {
      "taskId": "dGFza2lkZ29lc2hlcmUK",
      "resultCode": "Succeeded",
      "resultString": "[\"Mary Major", \"John Stiles\"]"
    }
  ]
}
```

Ejemplo de una función Lambda para Operaciones por lotes de S3

El siguiente ejemplo Python Lambda elimina un marcador de eliminación de un objeto versionado.

Como se muestra en el ejemplo, las claves de Operaciones por lotes de S3 están codificadas por URL. Para utilizar Amazon S3 con otros servicios de AWS, es importante que decodifique mediante la URL la clave que se pasa desde la herramienta de operaciones por lotes de S3.

```
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,
    }
```

Creación de un trabajo de Operaciones por lotes de S3 que invoca a una función Lambda

Al crear un trabajo de Operaciones por lotes de S3 para invocar a una función Lambda, debe proporcionar lo siguiente:

- El ARN de la función Lambda (que puede incluir el alias de la función o un número de versión específico)
- Un rol de IAM con permiso para invocar la función
- El parámetro de acción `LambdaInvokeFunction`

Para obtener más información acerca de cómo crear un trabajo de Operaciones por lotes de S3, consulte Creación de trabajos de operaciones por lotes de S3 y Operaciones compatibles con las operaciones por lotes de S3.

En el siguiente ejemplo, se crea un trabajo de la herramienta de operaciones por lotes de S3 que invoca una función de Lambda mediante 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"
```

Proporcionar información de tareas en manifiestos de Lambda

Cuando utiliza funciones de AWS Lambda con la herramienta de operaciones por lotes de S3, es posible que desee que se adjunten datos adicionales a cada tarea o clave en la que se opera. Por ejemplo, es posible que desee tener tanto una clave de objeto de origen como una nueva clave de objeto. Su función Lambda podría copiar la clave de origen en un nuevo bucket de S3 con un nuevo nombre. De forma predeterminada, operaciones por lotes de Amazon S3 le permite especificar solo el bucket de destino y una lista de claves de origen en el manifiesto de entrada para su trabajo. A continuación, se describe cómo puede incluir datos adicionales en su manifiesto para ejecutar funciones Lambda más complejas.

Para especificar parámetros por clave en el manifiesto de Operaciones por lotes de S3 para utilizarlos en el código de la función Lambda, use el siguiente formato JSON codificado mediante URL. El campo `key` se pasa a su función Lambda como si fuera una clave de objeto de Amazon S3. Pero la función Lambda puede interpretar que contiene otros valores o claves múltiples, como se muestra a continuación.

> **Note**
>
> El número máximo de caracteres para el campo `key` del manifiesto es 1024.

Example - Manifiesto que sustituye las "claves de Amazon S3" por cadenas JSON

La versión codificada mediante URL debe facilitarse a Operaciones por lotes de S3.

```
my-bucket,{"origKey": "object1key", "newKey": "newObject1Key"}
my-bucket,{"origKey": "object2key", "newKey": "newObject2Key"}
my-bucket,{"origKey": "object3key", "newKey": "newObject3Key"}
```

Example - Manifiesto codificado mediante URL

Esta versión codificada mediante URL debe facilitarse a Operaciones por lotes de S3. La versión no codificada en URL no funciona.

```
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 - Función Lambda con formato de manifiesto que escribe resultados en el informe del trabajo

Este ejemplo de manifiesto codificado en URL contiene claves de objeto delimitadas por barra vertical para que las analice la siguiente función de Lambda.

```
my-bucket,object1key%7Clower
my-bucket,object2key%7Cupper
my-bucket,object3key%7Creverse
my-bucket,object4key%7Cdelete
```

Esta función Lambda muestra cómo analizar una tarea delimitada por barra vertical que está codificada en el manifiesto de Operaciones por lotes de S3. La tarea indica qué operación de revisión se aplica al objeto especificado.

```
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,
    }
```

Tutorial de aprendizaje de operaciones por lotes de S3

En el siguiente tutorial se presentan procedimientos integrales completos para algunas tareas de operaciones por lotes con Lambda.

- Tutorial: videos de transcodificación por lotes con operaciones por lotes de S3, AWS Lambda, y AWS Elemental MediaConvert

Reemplazar todas las etiquetas de objeto

La operación Replace all object tags (Reemplazar todas las etiquetas de objeto) reemplaza las etiquetas de objeto de Amazon S3 en todos los objetos enumerados en el manifiesto. Las etiquetas de objetos de Amazon S3 son pares de cadenas clave-valor que se pueden utilizar para guardar metadatos de un objeto.

Para crear un trabajo Reemplazar todas las etiquetas de objeto, debe proporcionar un conjunto de etiquetas que desea aplicar. Operaciones por lotes de S3 aplica el mismo conjunto de etiquetas a cada objeto. El conjunto de etiquetas que proporcione reemplaza a los conjuntos de etiquetas que ya estén asociados con los objetos del manifiesto. Operaciones por lotes de S3 no permite agregar etiquetas a los objetos sin quitar las etiquetas existentes.

Si los objetos del manifiesto están en un bucket con control de versiones, puede aplicar un conjunto de etiquetas a versiones específicas de cada objeto. Para ello, especifique un ID de versión para cada objeto del manifiesto. Si no incluye el ID de versión de un objeto, Operaciones por lotes de S3 aplicará el conjunto de etiquetas a la versión más reciente del objeto.

Restricciones y limitaciones

- El rol de AWS Identity and Access Management (IAM) que especifique para ejecutar el trabajo de la herramienta de operaciones por lotes debe tener permisos para realizar la operación subyacente de Amazon S3 Reemplazar todas las etiquetas de objeto. Para obtener más información acerca de los permisos necesarios, consulte PutObjectTagging en la referencia de la API de Amazon Simple Storage Service.
- Operaciones por lotes de S3 utiliza la operación PutObjectTagging de Amazon S3 para aplicar etiquetas a cada objeto del manifiesto. Todas las restricciones y limitaciones que se aplican a la operación subyacente también se aplican a los trabajos de operaciones por lotes de S3.

Para obtener más información acerca del uso de la consola para crear trabajos, consulte Creación de un trabajo de operaciones por lotes de S3.

Para obtener más información acerca del etiquetado de objetos, consulte Categorización del almacenamiento mediante etiquetas en esta guía y PutObjectTagging, GetObjectTagging y DeleteObjectTagging en la Referencia de la API de Amazon Simple Storage Service.

Eliminar todas las etiquetas de objeto

La operación Delete all object tags (Eliminar todas las etiquetas de objeto) elimina todos los conjuntos de etiquetas de objeto de Amazon S3 asociados actualmente a los objetos enumerados en el manifiesto. Operaciones por lotes de S3 no admite la eliminación de etiquetas de objetos mientras mantiene otras etiquetas en su lugar.

Si los objetos del manifiesto están en un bucket versionado, puede eliminar los conjuntos de etiquetas de una versión específica de un objeto. Para ello, especifique un ID de versión para cada objeto del manifiesto. Si no incluye un ID de versión para un objeto, la herramienta de operaciones por lotes de S3 elimina el conjunto de etiquetas de la última versión de cada objeto.

Para obtener más información sobre los manifiestos de Operaciones por lotes, consulte Especificar un manifiesto.

> **Warning**
>
> Al ejecutar este trabajo, se eliminan todos los conjuntos de etiquetas de objeto en todos los objetos enumerados en el manifiesto.

Restricciones y limitaciones

- El rol de AWS Identity and Access Management (IAM) que especifique para ejecutar el trabajo debe tener permisos para realizar la operación subyacente de etiquetado de objetos de Amazon S3 Delete. Para obtener más información, consulte DeleteObjectTagging en la Referencia de API de Amazon Simple Storage Service.
- Operaciones por lotes de S3 utiliza la operación DeleteObjectTagging de Amazon S3 para eliminar los conjuntos de etiquetas de todos los objetos del manifiesto. Todas las restricciones y limitaciones que se aplican a la operación subyacente también se aplican a los trabajos de operaciones por lotes de S3.

Para obtener más información acerca de la creación de trabajos, consulte Creación de trabajos de operaciones por lotes de S3.

Para obtener más información sobre el etiquetado de objetos, consulte Reemplazar todas las etiquetas de objeto en esta guía y PutObjectTagging, GetObjectTaggingy DeleteObjectTagging en la Referencia de API de Amazon Simple Storage Service.

Reemplazar la lista de control de acceso

La operación Reemplazar lista de control de acceso (ACL) reemplaza las listas de control de acceso (ACL) de Amazon S3 para cada objeto que se muestra en el manifiesto. Con las ACL, es posible determinar quién puede tener acceso a un objeto y qué acciones puede realizar.

Operaciones por lotes de S3 permite utilizar ACL personalizadas que los usuarios pueden definir y ACL predefinidas que están incluidas en Amazon S3 con un conjunto de permisos de acceso preestablecidos.

Si los objetos del manifiesto están en un bucket con control de versiones, puede aplicar las ACL a versiones específicas de cada objeto. Para ello, especifique un ID de versión para cada objeto del manifiesto. Si no incluye el ID de versión de un objeto, Operaciones por lotes de S3 aplicará la ACL a la versión más reciente del objeto.

Para obtener más información acerca de las ACL en Amazon S3, consul3, Información general de las Listas de control de acceso (ACL).

S3 Block Public Access

Si desea limitar el acceso público en todos los objetos de un bucket, debe utilizar el bloqueo de acceso público de Amazon S3 en lugar de las operaciones por lotes de S3. Con el bloqueo de acceso público, se puede limitar el acceso público en cada bucket o en toda la cuenta mediante una única y sencilla operación que surte efecto rápidamente. Esta opción es más adecuada cuando el objetivo es controlar el acceso público en todos los objetos de un bucket o una cuenta. Utilice Operaciones por lotes de S3 cuando necesite aplicar una ACL personalizada a cada uno de los objetos del manifiesto. Para obtener más información acerca del bloqueo de acceso público de S3, consulte Bloquear el acceso público a su almacenamiento de Amazon S3.

Propiedad de objetos de S3

Si los objetos del manifiesto se encuentran en un bucket que utiliza la configuración de propietario del bucket obligatorio de Object Ownership, la operación Reemplazar lista de control de acceso

(ACL) solo puede especificar las ACL de objetos que conceden control total al propietario del bucket. La operación no puede conceder permisos de ACL de objeto a otras Cuentas de AWS o grupos. Para obtener más información, consulte Control de la propiedad de los objetos y desactivación de las ACL del bucket.

Restricciones y limitaciones

- El rol que especifique para ejecutar el trabajo “Reemplazar lista de control de acceso” debe tener permisos para realizar la operación subyacente `PutObjectAcl` de Amazon S3. Para obtener más información acerca de los permisos necesarios, consulte PutObjectAcl en la Referencia de la API de Amazon Simple Storage Service.
- Las operaciones por lotes de S3 utilizan la operación `PutObjectAcl` de Amazon S3 para aplicar la ACL especificada a cada objeto del manifiesto. Por lo tanto, todas las restricciones y limitaciones que se aplican a la operación PutObjectAcl subyacente también se aplican a los trabajos de lista de control de acceso S3 Operaciones por lotes Reemplazar.

Restaurara objetos con Operaciones por lotes

La operación Restaurar inicia solicitudes de restauración para los objetos de Amazon S3 archivados que se enumeran en el manifiesto. Los siguientes objetos archivados deben restaurarse para poder acceder a ellos en tiempo real:

- Objetos archivados en las clases de almacenamiento S3 Glacier Flexible Retrieval o S3 Glacier Deep Archive
- Objetos archivados a través de la clase de almacenamiento S3 Intelligent-Tiering en los niveles de acceso a archivos o acceso profundo a archivos

El uso de una operación de Iniciar objeto de restauración de S3 en el trabajo de operaciones por lotes de S3 da como resultado una solicitud de restauración para cada objeto especificado en el manifiesto.

> **Important**
>
> El trabajo Iniciar restauración de objeto S3 solo inicia la solicitud para restaurar objetos. Operaciones por lotes de S3 notifica que el trabajo está completado para cada objeto después de que se haya iniciado la solicitud para ese objeto. Amazon S3 no actualiza el trabajo ni le notifica cuando se han restaurado los objetos. No obstante, puede utilizar

las notificaciones de eventos de S3 para recibir notificaciones cuando los objetos estén disponibles en Amazon S3. Para obtener más información, consulte Notificaciones de eventos de Amazon S3.

Al crear un trabajo de Iniciar restauración de objetos de S3, están disponibles los siguientes argumentos:

ExpirationInDays

Este argumento especifica cuánto tiempo el objeto S3 Glacier Flexible Retrieval o S3 Glacier Deep Archive permanece disponible en Amazon S3. Los trabajos de Iniciar restauración de objetos que se dirigen a objetos de S3 Glacier Flexible Retrieval y S3 Glacier Deep Archive requieren que establezca `ExpirationInDays` a 1 o superior.

> **Important**
>
> No establezca `ExpirationInDays` al crear trabajos de operación de Iniciar restauración de objetos de S3 que se dirijan a objetos de nivel Acceso a archivos y Acceso a archivos profundo de S3 Intelligent-Tiering. Los objetos de los niveles de acceso a archivos de S3 Intelligent-Tiering no están sujetos a caducidad de la restauración, por lo que especificar `ExpirationInDays` da como resultado un error en la solicitud de restauración.

GlacierJobTier

Amazon S3 puede restaurar objetos mediante uno de tres niveles de recuperación: `EXPEDITED`, `STANDARD` y `BULK`. Sin embargo, la característica de Operaciones por lotes de S3 solo admite los niveles de recuperación `STANDARD`. Para obtener más información sobre las diferencias entre los niveles de recuperación, consulte Opciones de recuperación de archivos.

Para obtener más información sobre los precios de cada nivel, consulte la sección Solicitudes y recuperaciones de datos en la página Precios de Amazon S3.

Diferencias al restaurar desde S3 Glacier y S3 Intelligent-Tiering

La restauración de archivos guardados desde las clases de almacenamiento S3 Glacier Flexible Retrieval o S3 Glacier Deep Archive difiere de la restauración de archivos de la clase de almacenamiento de S3 Intelligent-Tiering en los niveles Archive Access o Deep Archive Access.

- Cuando se restaura desde S3 Glacier Flexible Retrieval o S3 Glacier Deep Archive, se crea una copia temporal del objeto. Amazon S3 elimina esta copia una vez transcurrido el valor especificado en el argumento `ExpirationInDays`. Después de eliminar esta copia temporal, debe enviar una solicitud de restauración adicional para tener acceso al objeto.
- No especifique el argumento `ExpirationInDays` al restaurar objetos archivados de S3 Intelligent-Tiering. Cuando se restaura un objeto desde los niveles Acceso a archivos o Acceso a archivos profundo de S3 Intelligent-Tiering, el objeto vuelve al nivel Acceso frecuente de S3 Intelligent-Tiering. Tras un mínimo de 90 días consecutivos sin acceso, el objeto pasa automáticamente al nivel Acceso a archivos. Tras un mínimo de 180 días consecutivos sin acceso, el objeto pasa automáticamente al nivel de Acceso a archivos profundo.
- Los trabajos de Operaciones por lotes pueden operar en objetos de clase de almacenamiento de S3 Glacier Flexible Retrieval y S3 Glacier Deep Archive o en objetos de nivel de almacenamiento Acceso a archivos o Acceso a archivos profundo de S3 Intelligent-Tiering. Operaciones por lotes no pueden funcionar con ambos tipos de objetos archivados en el mismo trabajo. Para restaurar objetos de ambos tipos, debe crear trabajos de operaciones por lotes independientes.

Solapar restauraciones

Si un trabajo Iniciar restauración de objetos de S3 intenta restaurar un objeto que ya se está restaurando, las operaciones por lotes de S3 se comportarán del modo siguiente:

La operación de restauración se realizará correctamente en el objeto si se da alguna de las siguientes condiciones:

- En comparación con la solicitud de restauración ya en curso, el valor `ExpirationInDays` de este trabajo es el mismo y su valor `GlacierJobTier` es más rápido.
- La solicitud de restauración anterior ya se ha completado y el objeto está disponible en este momento. En este caso, Operaciones por lotes actualiza la fecha de caducidad del objeto restaurado para que coincida con el valor `ExpirationInDays` especificado en la solicitud de restauración en curso.

La operación de restauración del objeto no se realizará correctamente si se da alguna de las siguientes condiciones:

- La solicitud de restauración ya está en curso pero aún no se ha completado y la duración de restauración de este trabajo (especificada mediante el valor `ExpirationInDays`) es diferente de la duración de la restauración especificada en la solicitud de restauración en curso.

- El nivel de restauración de este trabajo (especificado mediante el valor `GlacierJobTier`) es igual o más lento que el que se especificó en la solicitud de restauración en curso.

Limitaciones

Los trabajos de Iniciar restauración de objetos S3 tienen las siguientes limitaciones:

- Debe crear el trabajo en la misma región que los objetos archivados.
- Las operaciones por lotes de S3 no admiten el nivel de recuperación `EXPEDITED`.

Para obtener más información acerca de la restauración de objetos, consulte Restauración de un objeto archivado.

Retención de bloqueo de objetos S3

La operación de Object Lock retention (Retención de bloqueo de objetos) permite aplicar fechas de retención para los objetos mediante el modo de gobierno o el modo de conformidad. Estos modos de retención aplican diferentes niveles de protección. Puede aplicar cualquiera de los modos de retención a cualquier versión de objeto. Las fechas de retención, al igual que las retenciones legales, impiden que un objeto se sobrescriba o elimine. Amazon S3 almacena la fecha de finalización de la retención que se especifica en los metadatos del objeto y protege la versión especificada de la versión del objeto hasta que vence el período de retención.

Puede utilizar Operaciones por lotes de S3 con Bloqueo de objetos para administrar las fechas de retención de muchos objetos de Amazon S3 al mismo tiempo. Especifique la lista de objetos de destino en el manifiesto y envíela a Operaciones por lotes para su finalización. Para obtener más información, consulte de Bloqueo de objetos de S the section called “Periodos de retención”.

El trabajo de Operaciones por lotes de S3 con las fechas de retención se ejecuta hasta que termina, se cancela o llega a un estado de error. Debe utilizar Operaciones por lotes de S3 y la retención de Bloqueo de objetos de S3 cuando desee añadir, cambiar o quitar la fecha de retención de muchos objetos con una sola solicitud.

Operaciones por lotes comprueba que el Bloqueo de objetos está activado en el bucket antes de procesar cualquier clave en el manifiesto. Para realizar las operaciones y la validación, Operaciones por lotes necesita los permisos `s3:GetBucketObjectLockConfiguration` y `s3:PutObjectRetention` en un rol de IAM para permitir que Operaciones por lotes llame al Bloqueo de objetos en su nombre. Para obtener más información, consulte the section called “Consideraciones sobre el bloqueo de objetos”.

Para obtener información sobre el uso de esta operación con la API de REST, consulte `S3PutObjectRetention` en la operación CreateJob de la referencia de la API de Amazon Simple Storage Service.

Para obtener un ejemplo AWS Command Line Interface del uso de esta operación, consulte the section called “Uso de Operaciones por lotes con retención de Bloqueo de objetos”. Para ver un ejemplo de AWS SDK for Java, consulte the section called “Uso de Operaciones por lotes con retención de Bloqueo de objetos”.

Restricciones y limitaciones

- Operaciones por lotes de S3 no realiza ningún cambio en el nivel del bucket.
- El control de versiones y el Bloqueo de objetos de S3 deben configurarse en el bucket donde se realiza el trabajo.
- Todos los objetos enumerados en el manifiesto deben estar en el mismo bucket.
- La operación funciona en la versión más reciente del objeto a menos que se especifique explícitamente una versión en el manifiesto.
- Necesita permisos `s3:PutObjectRetention` en el rol de IAM para usar esto.
- `s3:GetBucketObjectLockConfiguration`Se necesita el permiso de IAM para confirmar que el Bloqueo de objetos está activado para el bucket de S3.
- Solo se puede ampliar el período de retención de objetos con fechas de retención de modo `COMPLIANCE` aplicadas y no se puede acortar.

Bloqueo de objetos de retención legal en S3

La operación de Object Lock legal hold (Retención legal de bloqueo de objetos) permite colocar una retención legal en una versión de objeto. Al igual que un periodo de retención, la retención legal impide que se sobrescriba o elimine una versión de un objeto. Sin embargo, una retención legal no tiene asociado un periodo de retención y sigue vigente hasta que se elimine.

Puede utilizar Operaciones por lotes de S3 con Bloqueo de objetos para agregar retenciones legales a muchos objetos de Amazon S3 al mismo tiempo. Para ello, obtenga una lista de los objetos de destino en su manifiesto y envíela a Operaciones por lotes. El trabajo de Operaciones por lotes de S3 con una retención legal de Bloqueo de objetos se ejecuta hasta que termina, se cancela o llega a un estado de error.

Operaciones por lotes de S3 comprueba que el Bloqueo de objetos está activado en el bucket de S3 antes de procesar cualquier clave en el manifiesto. Para realizar las operaciones del objeto y la validación a nivel de bucket, Operaciones por lotes de S3 necesita `s3:PutObjectLegalHold` y `s3:GetBucketObjectLockConfiguration` en un rol de IAM que permita a Operaciones por lotes de S3 llamar al Bloqueo de objetos de S3 en su nombre.

Al crear el trabajo de Operaciones por lotes de S3 para eliminar la retención legal, solo tiene que especificar Off (Desactivado) como estado de retención legal. Para obtener más información, consulte the section called “Consideraciones sobre el bloqueo de objetos”.

Para obtener información sobre el uso de esta operación con la API de REST, consulte `S3PutObjectLegalHold` en la operación CreateJob de la referencia de la API de Amazon Simple Storage Service.

Para obtener un ejemplo de uso de esta operación, consulte Uso de AWS SDK para Java.

Restricciones y limitaciones

- Operaciones por lotes de S3 no realiza ningún cambio en el nivel del bucket.
- Todos los objetos enumerados en el manifiesto deben estar en el mismo bucket.
- El control de versiones y el Bloqueo de objetos de S3 deben configurarse en el bucket donde se realiza el trabajo.
- La operación funciona en la versión más reciente del objeto a menos que se especifique explícitamente una versión en el manifiesto.
- `s3:PutObjectLegalHold`Se necesita el permiso en el rol de IAM para añadir o quitar la retención legal de los objetos.
- `s3:GetBucketObjectLockConfiguration`Se necesita el permiso de IAM para confirmar que Bloqueo de objetos de S3 está activado para el bucket de S3.

- Copia de objetos
- Invocar a la función AWS Lambda
- Reemplazar todas las etiquetas de objeto
- Eliminar todas las etiquetas de objeto
- Reemplazar la lista de control de acceso
- Restaurara objetos con Operaciones por lotes

- Retención de bloqueo de objetos S3
- Bloqueo de objetos de retención legal en S3
- Replicación de objetos existentes con replicación por lotes de S3

Administración de trabajos de operaciones por lotes de S3

Amazon S3 proporciona un conjunto de herramientas sólido que lo ayudarán a administrar los trabajos de las operaciones por lotes de S3 una vez que los haya creado. En esta sección, se describen las operaciones que puede utilizar para administrar sus trabajos y realizar un seguimiento de ellos con la AWS Management Console, la AWS CLI, los AWS SDK o la API REST.

Temas

- Uso de la consola de Simple Storage Service (Amazon S3) para administrar sus trabajos de operaciones por lotes de S3
- Mostrar trabajos
- Consultar detalles del trabajo
- Asignar prioridad a los trabajos

Uso de la consola de Simple Storage Service (Amazon S3) para administrar sus trabajos de operaciones por lotes de S3

Puede administrar sus trabajos de operaciones por lotes de S3 con la consola. Por ejemplo, puede hacer lo siguiente:

- Visualización de trabajos activos y en cola
- Cambio de la prioridad de un trabajo
- Confirmación y ejecución de un trabajo
- Clonación de un trabajo
- Cancelación de un trabajo

Pasos para administrar las operaciones por lotes con la consola

1. Inicie sesión AWS Management Console Management Console y abra la consola de Amazon S3 en https://console.aws.amazon.com/s3/.

2. En el panel de navegación izquierdo, elija Batch Operations (Operaciones por lote).
3. Elija el trabajo específico que desearía administrar.

Mostrar trabajos

Puede recuperar una lista de los trabajos de Operaciones por lotes de S3. Esta lista incluirá los trabajos que aún no han finalizado y los que finalizaron en los últimos 90 días. La lista contiene información de cada trabajo, como el ID, la descripción, la prioridad, el estado actual y el número de trabajos realizados con éxito y que han dado error. Puede filtrar la lista de trabajos por estado. Si obtiene la lista de trabajos a través de la consola, también podrá buscar los trabajos por descripción o ID y filtrarlos por Región de AWS.

Obtener una lista de trabajos activos y finalizados

En el ejemplo siguiente de AWS CLI se obtiene una lista de los trabajos `Active` y `Complete`.

```
aws s3control list-jobs \
    --region us-west-2 \
    --account-id acct-id \
    --job-statuses '["Active","Complete"]' \
    --max-results 20
```

Para obtener más información y ejemplos, consulte list-jobs en la referencia de comandos de AWS CLI.

Consultar detalles del trabajo

Si desea más información sobre un trabajo que puede recuperar creando una lista de trabajos, puede ver todos los detalles de un solo trabajo. Puede ver los detalles de los trabajos que aún no han finalizado y los que finalizaron en los últimos 90 días. Además de la información que aparece en la lista de trabajos, los detalles de los trabajos contienen otros elementos como los siguientes:

- parámetros de operación
- detalles sobre el manifiesto
- Información sobre el reporte de finalización (si configuró uno cuando creó el trabajo)
- nombre de recurso de Amazon (ARN) del rol de usuario que asignó para ejecutar el trabajo

Al ver los detalles de un trabajo individual, puede acceder a la configuración completa del trabajo. Para ver los detalles de un trabajo, puede utilizar la consola de Amazon S3 o la AWS Command Line Interface (AWS CLI).

Obtención de la descripción de un trabajo de Operaciones por lotes de S3 en la consola de Amazon S3

Para ver la descripción de un trabajo de Operaciones por lotes mediante la consola

1. Inicie sesión AWS Management Console Management Console y abra la consola de Amazon S3 en https://console.aws.amazon.com/s3/.
2. En el panel de navegación izquierdo, elija Batch Operations (Operaciones por lote).
3. Seleccione el ID del trabajo en cuestión para ver sus detalles.

Obtención de la descripción de un trabajo de Operaciones por lotes de S3 en la AWS CLI

En el siguiente ejemplo, se obtiene la descripción de un trabajo de Operaciones por lotes de S3 mediante la AWS CLI. Para utilizar el comando de ejemplo siguiente, sustituya *`user input placeholders`* con su información.

```
aws s3control describe-job \
--region us-west-2 \
--account-id acct-id \
--job-id 00e123a4-c0d8-41f4-a0eb-b46f9ba5b07c
```

Para obtener más información y ejemplos, consulte describe-job en la referencia de comandos de AWS CLI.

Asignar prioridad a los trabajos

Puede asignar a cada trabajo una prioridad numérica, que puede ser cualquier entero positivo. Operaciones por lotes de S3 prioriza los trabajos de acuerdo con la prioridad asignada. Los trabajos con una prioridad mayor (o un valor numérico más elevado en el parámetro priority) se evalúan en primer lugar. La prioridad se determina en orden descendente. Por ejemplo, una cola de trabajos con una prioridad 10, tendrá mayor preferencia de programación que una cola de trabajos cuyo valor de prioridad sea 1.

La prioridad de un trabajo se puede modificar mientras está en ejecución. Si se envía un nuevo trabajo con una prioridad mayor mientras hay otro trabajo en ejecución, el trabajo con menor prioridad se detendrá para permitir que se ejecute el de mayor prioridad.

Cambiar la prioridad del trabajo no afecta a la velocidad de procesamiento de trabajos.

> **Note**
>
> Operaciones por lotes de S3 respeta las prioridades de los trabajos en la medida de lo posible. Aunque, por lo general, los trabajos con mayor prioridad prevalecerán sobre los trabajos con una prioridad más baja, Amazon S3 no garantiza un orden estricto de los trabajos.

Uso de la consola de S3

Cómo actualizar la prioridad del trabajo en la AWS Management Console

1. Inicie sesión AWS Management Console Management Console y abra la consola de Amazon S3 en https://console.aws.amazon.com/s3/.
2. En el panel de navegación izquierdo, elija Batch Operations (Operaciones por lote).
3. Seleccione el trabajo específico que desearía administrar.
4. Elija Actions (Acciones). En la lista desplegable, elija Update priority (Actualizar la prioridad).

Utilización del AWS CLI

En el ejemplo siguiente se actualiza la prioridad del trabajo mediante la AWS CLI. Un número más alto indica una prioridad de ejecución más alta.

```
aws s3control update-job-priority \
    --region us-west-2 \
    --account-id acct-id \
    --priority 98 \
    --job-id 00e123a4-c0d8-41f4-a0eb-b46f9ba5b07c
```

Mediante AWS SDK for Java

En el siguiente ejemplo, se actualiza la prioridad de un trabajo de la herramienta de operaciones por lotes de S3 mediante AWS SDK for Java.

Para obtener más información acerca de la prioridad de un trabajo, consulte Asignar prioridad a los trabajos.

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();
        }
    }
```

```
}
```

Seguimiento del estado del trabajo e informes de finalización

Con las operaciones por lotes de S3, puede ver y actualizar el estado del trabajo, agregar notificaciones y registros, realizar un seguimiento de los errores del trabajo y generar informes de finalización.

Temas

Estados de los trabajos

Cuando se crea y ejecuta un trabajo, este pasa por una serie de estados. En la siguiente tabla se describen los estados y las posibles transiciones entre ellos.

Estado	Descripción	Transiciones
`New`	Cuando se crean, los trabajos tienen el estado `New`.	Un trabajo pasa automáticamente al estado `Preparing` cuando Amazon S3 comienza a procesar el objeto del manifiesto.
`Preparing`	Amazon S3 está procesando el objeto del manifiesto y otros parámetros del trabajo para configurar y ejecutar el trabajo.	Los trabajos pasan automáticamente al estado `Ready` cuando Amazon S3 termina de procesar el manifiesto y otros parámetros. En ese

Estado	Descripción	Transiciones
		momento, el trabajo está listo para ejecutar la operación especificada sobre los objetos que aparecen en el manifiesto. Si el trabajo necesita confirmación antes de ejecutarse (por ejemplo: cuando se crea un trabajo utilizando la consola de Amazon S3), el trabajo pasa de `Preparing` a `Suspended`. El trabajo permanecerá con el estado `Suspended` hasta que confirme que desea ejecutarlo.
`Suspended`	El trabajo necesita confirmación, pero aún no ha confirmado que desea ejecutarlo. Los únicos trabajos que necesitan confirmación son los que se crean con la consola de Amazon S3. Los trabajos que se crean con la consola pasan al estado `Suspended` inmediatamente después de `Preparing`. Una vez que confirme que desea ejecutar el trabajo y este pase al estado `Ready`, el trabajo nunca volverá a tener el estado `Suspended`.	Cuando confirme que desea ejecutar el trabajo, el estado cambiará a `Ready`.

Estado	Descripción	Transiciones
`Ready`	Amazon S3 está listo para comenzar a ejecutar las operaciones solicitadas en los objetos.	Los trabajos pasan automáticamente al estado `Active` cuando Amazon S3 comienza a ejecutarlos. El tiempo que un trabajo permanece en el estado `Ready` depende de si hay trabajos con una prioridad mayor que ya estén en ejecución y de lo que esos trabajos tarden en completarse.
`Active`	Amazon S3 está ejecutando la operación solicitada en los objetos que aparecen en el manifiesto. Mientras un trabajo está en el estado `Active`, puede monitorear su progreso mediante la consola de Amazon S3 o la operación `DescribeJob` a través de la API REST, la AWS CLI o los SDK de AWS.	Los trabajos pasan al estado `Active` cuando ya no hay operaciones de objetos en ejecución. Esta transición se realiza automáticamente; por ejemplo, cuando un trabajo se completa correctamente o encuentra un error. La transición también puede producirse como consecuencia de la acción de un usuario; por ejemplo, cuando un usuario cancela un trabajo. El estado al que pasa el trabajo depende del motivo de la transición.
`Pausing`	El trabajo tenía otro estado y ha pasado a `Paused`.	Los trabajos pasan automáticamente al estado `Paused` cuando finaliza la etapa de `Pausing`.

Estado	Descripción	Transiciones
`Paused`	Los trabajos adoptan el estado `Paused` si se envía un trabajo con una prioridad mayor mientras el trabajo actual está en ejecución.	Los trabajos con el estado `Paused` pasan automáticamente a `Active` cuando otros trabajos de mayor prioridad que impiden su ejecución se completan, encuentran un error o se suspenden.
`Complete`	El trabajo ha terminado de ejecutar la operación solicitada en todos los objetos del manifiesto. La operación puede haberse ejecutado correctamente o con errores en cada objeto. Si configuró el trabajo para que se generara un informe de finalización, dicho informe estará disponible tan pronto como el trabajo adopte el estado `Complete`.	`Complete` es un estado terminal. Cuando un trabajo alcanza el estado `Complete`, ya no adopta otros estados.
`Cancelling`	El trabajo ha adoptado el estado `Cancelled` .	Los trabajos pasan automáticamente al estado `Cancelled` cuando finaliza la etapa de `Cancelling` .
`Cancelled`	Se ha solicitado la cancelación del trabajo y las operaciones por lotes de S3 lo ha cancelado con éxito. El trabajo ya no enviará ninguna solicitud nueva a Amazon S3.	`Cancelled` es un estado terminal. Cuando un trabajo alcanza el estado `Cancelled` , ya no adopta ningún otro estado.

Estado	Descripción	Transiciones
`Failing`	El trabajo ha adoptado el estado `Failed`.	Los trabajos pasan automáticamente al estado `Failed` cuando finaliza la etapa de `Failing`.
`Failed`	Se ha producido un error en el trabajo y ya no está en ejecución. Para obtener más información acerca de los errores de los trabajos, consulte Realizar un seguimiento de los errores de los trabajos.	`Failed` es un estado terminal. Cuando un trabajo alcanza el estado `Failed`, ya no adopta ningún otro estado.

Actualización del estado del trabajo

Mediante los siguientes ejemplos de la AWS CLI y el SDK para Java, se actualiza el estado de un trabajo de la herramienta de operaciones por lotes. Para obtener más información acerca del uso de la consola de S3 para administrar trabajos de las operaciones por lotes, consulte Uso de la consola de Simple Storage Service (Amazon S3) para administrar sus trabajos de operaciones por lotes de S3.

Mediante AWS CLI

- Si no especificó el parámetro `--no-confirmation-required` en el ejemplo anterior de `create-job`, el trabajo permanece suspendido hasta que lo confirme estableciendo su estado en `Ready`. A continuación, Amazon S3 hace que el trabajo sea apto para su ejecución.

```
aws s3control update-job-status \
    --region us-west-2 \
    --account-id 181572960644 \
    --job-id 00e123a4-c0d8-41f4-a0eb-b46f9ba5b07c \
    --requested-job-status 'Ready'
```

- Para cancelar el trabajo, establezca su estado en `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
```

Uso de AWS SDK para Java

En el siguiente ejemplo, se actualiza el estado de un trabajo de la herramienta de operaciones por lotes de S3 mediante AWS SDK for Java.

Para obtener más información acerca del estado de un trabajo, consulte Seguimiento del estado del trabajo e informes de finalización.

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();
        }
    }
}
```

Notificaciones y registro

Además de solicitar informes de finalización, también puede obtener, revisar y auditar la actividad de las operaciones por lotes mediante AWS CloudTrail. Como Operaciones por lotes utiliza las API de Amazon S3 existentes para realizar tareas, esas tareas también emiten los mismos eventos que si se las llamara directamente. Así, puede realizar un seguimiento y registrar el progreso de su trabajo y todas sus tareas mediante las mismas herramientas y procesos de notificación, registro y auditoría que ya utiliza con Amazon S3. Para obtener más información, consulte los ejemplos de las siguientes secciones.

> **Note**
>
> Las operaciones por lote de Amazon S3 generan eventos de administración y datos en CloudTrail durante la ejecución del trabajo. El volumen de estos eventos se escala con el número de claves en el manifiesto de cada trabajo. Consulte la página de Precios de CloudTrail para obtener más información, que incluye ejemplos de cómo cambian los precios en función de la cantidad de registros de seguimiento que haya configurado en la cuenta. Para obtener información acerca de cómo configurar y registrar eventos para que se ajusten a sus necesidades, consulte Crear su primer registro de seguimiento en la Guía del usuario de AWS CloudTrail.

Para obtener más información acerca de los eventos de Amazon S3, consulte Notificaciones de eventos de Amazon S3.

Realizar un seguimiento de los errores de los trabajos

Si un trabajo de Operaciones por lotes de S3 se encuentra con un problema que impide su correcta ejecución (por ejemplo: no se puede leer el manifiesto especificado), se produce un error. Cuando un trabajo falla, genera uno o más códigos de error o motivos de error. Operaciones por lotes de S3 almacena los códigos de error y los motivos con el trabajo para que pueda consultarlos solicitando los detalles del trabajo. Si solicitó un informe de finalización para el trabajo, los códigos de error y los motivos también aparecen aquí.

Para impedir que los trabajos ejecuten un número elevado de operaciones incorrectamente, Amazon S3 impone un umbral de errores de tareas en cada trabajo de Operaciones por lotes de S3. Una vez que un trabajo ha ejecutado al menos 1000 tareas, Amazon S3 monitoriza la tasa de errores de tareas. Si en algún momento la tasa de errores (el número de tareas con error en comparación con el número total de tareas ejecutadas) supera el 50 %, el trabajo genera un error. Si el trabajo deja de ejecutarse porque se ha superado el umbral de errores de tareas, es posible identificar la causa de dichos errores. Por ejemplo, podría ocurrir que, por accidente, haya incluido en el manifiesto algunos objetos que no existen en el bucket especificado. Después de solucionar los errores, podrá volver a enviar el trabajo.

> **Note**
>
> Operaciones por lotes de S3 se ejecuta de forma asincrónica y las tareas no tienen que ejecutarse necesariamente en el mismo orden en el que los objetos aparecen en el manifiesto. Por tanto, no puede utilizar el orden del manifiesto para determinar qué tareas de los objetos se ejecutaron o no correctamente. En su lugar, puede examinar el informe de finalización del trabajo (si solicitó uno) o ver los registros de eventos de AWS CloudTrail para tratar de determinar el origen de los errores.

Informes de finalización

Al crear un trabajo, puede solicitar un informe de finalización. Si Operaciones por lotes de S3 invoca correctamente al menos una tarea, Amazon S3 generará un informe de finalización cuando las tareas terminen de ejecutarse, encuentren algún error o se cancelen. Puede configurar el informe de finalización para incluir todas las tareas o solo las tareas con error.

El informe de finalización incluye la configuración del trabajo, el estado y la información de cada tarea, incluidas la clave y la versión del objeto, el estado, códigos de error y descripciones de los errores. Los informes de finalización constituyen un mecanismo sencillo para ver los resultados de las tareas en un formato unificado sin necesidad de realizar ninguna configuración adicional. Los informes de finalización se cifran con claves administradas por Amazon S3 (SSE-S3). Para ver un ejemplo de un informe de finalización, consulte Ejemplos: informes de finalización de las operaciones por lotes de S3.

Aunque no se configure un informe de finalización, es posible monitorear y auditar el trabajo y las tareas con CloudTrail y Amazon CloudWatch. Para obtener más información, consulte la siguiente sección.

Temas

- Ejemplos: seguimiento de un trabajo de las operaciones por lotes de S3 en Amazon EventBridge a través de AWS CloudTrail
- Ejemplos: informes de finalización de las operaciones por lotes de S3

Ejemplos: seguimiento de un trabajo de las operaciones por lotes de S3 en Amazon EventBridge a través de AWS CloudTrail

La actividad de los trabajos de las operaciones por lotes en Amazon S3 se registra como eventos en AWS CloudTrail. Puede crear una regla personalizada en Amazon EventBridge y enviar estos eventos al recurso de notificación de destino que desee, como Amazon Simple Notification Service (Amazon SNS).

> **Note**
>
> Amazon EventBridge es la forma preferida de administrar sus eventos. Amazon CloudWatch Events y EventBridge son el mismo servicio subyacente y la misma API, pero EventBridge ofrece más características. Los cambios que realice en CloudWatch o EventBridge aparecerán en cada consola. Para obtener más información, consulte la Guía del usuario de Amazon EventBridge.

Ejemplos de seguimiento

- Eventos de operaciones por lotes de S3 registrados en CloudTrail
- Regla de EventBridge para el seguimiento de eventos de trabajo de operaciones por lotes de S3

Eventos de operaciones por lotes de S3 registrados en CloudTrail

Cuando se crea un trabajo de Operaciones por lotes, se registra como un evento `JobCreated` en CloudTrail. A medida que se ejecuta el trabajo, cambia de estado durante el procesamiento y se registran otros eventos `JobStatusChanged` en CloudTrail. Puede ver estos eventos en la consola de CloudTrail. Para obtener más información acerca de CloudTrail, consulte la Guía del usuario de AWS CloudTrail.

> **Note**
>
> Solo los eventos `status-change` del trabajo de Operaciones por lotes de S3 se registran en CloudTrail.

Example Evento de finalización del trabajo de operaciones por lotes de S3 registrado por 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": []
    }
}
```

Regla de EventBridge para el seguimiento de eventos de trabajo de operaciones por lotes de S3

En el siguiente ejemplo, se muestra cómo crear una regla en Amazon EventBridge para capturar los eventos de la herramienta de operaciones por lotes de S3 registrados por AWS CloudTrail en un destino de su elección.

Para ello, cree una regla siguiendo todos los pasos de Creación de reglas de EventBridge que reaccionen a los eventos. Puede pegar la siguiente política de patrones de eventos personalizados de Operaciones por lotes de S3 cuando corresponda y elegir el servicio de destino que desee.

Política de patrones de eventos personalizados de Operaciones por lotes de S3

```
{
    "source": [
        "aws.s3"
    ],
    "detail-type": [
        "AWS Service Event via CloudTrail"
    ],
    "detail": {
        "eventSource": [
            "s3.amazonaws.com"
        ],
        "eventName": [
            "JobCreated",
            "JobStatusChanged"
        ]
    }
}
```

Los siguientes ejemplos son dos eventos de Operaciones por lotes que se enviaron a Amazon Simple Queue Service (Amazon SQS) desde una regla de eventos de EventBridge. Un trabajo de Operaciones por lotes pasa por muchos estados diferentes durante el procesamiento (`New`, `Preparing`, `Active`, etc.), por lo que puede esperar recibir varios mensajes para cada trabajo.

Example Ejemplo de evento 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 Evento de finalización de trabajo 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": []
    }
  }
}
```

Ejemplos: informes de finalización de las operaciones por lotes de S3

Al crear un trabajo de Operaciones por lotes de S3, puede solicitar un informe de finalización para todas las tareas o solo para las tareas que no se realicen. Siempre que se haya invocado correctamente al menos una tarea, Operaciones por lotes de S3 genera un informe para los trabajos que se han completado, que han fallado o que se han cancelado.

El informe de finalización contiene información adicional sobre cada tarea, incluidas la clave y la versión del objeto, el estado, códigos de error y descripciones de los errores. La descripción de los

errores de cada tarea fallida se puede utilizar para diagnosticar problemas que surgen durante la creación de trabajos, como los permisos.

> **Note**
>
> Los informes de finalización siempre se cifran con claves administradas por Amazon S3 (SSE-S3).

Example archivo de resultados de manifiesto de nivel superior

El archivo `manifest.json` de nivel superior contiene las ubicaciones de cada informe correcto y (si el trabajo tuvo algún error) la ubicación de los informes con errores, como se muestra en el siguiente ejemplo.

```
{
    "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 informes de tareas con errores

Los informes de tareas fallidas contienen la siguiente información para todas las tareas fallidas :

- `Bucket`
- `Key`
- `VersionId`
- `TaskStatus`
- `ErrorCode`
- `HTTPStatusCode`
- `ResultMessage`

El siguiente informe de ejemplo muestra un caso en el que la función de AWS Lambda agotó el tiempo de espera, provocando que se sobrepasara el umbral de errores. Esto hizo que se marcara como `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 informes de tareas correctas

Los informes de tareas correctas incluyen lo siguiente para las tareas correctas:

- `Bucket`
- `Key`

- VersionId
- TaskStatus
- ErrorCode
- HTTPStatusCode
- ResultMessage

En el siguiente ejemplo, la función de Lambda copió correctamente el objeto de Amazon S3 en otro bucket. La respuesta de Amazon S3 devuelta se envía a Operaciones por lotes de S3 y, a continuación, se escribe en el informe de finalización 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'}}}"
```

Controlar el acceso y etiquetar trabajos usando etiquetas

Para etiquetar y controlar el acceso a los trabajos de Operaciones por lotes de S3, puede añadir etiquetas. Las etiquetas se pueden utilizar para identificar quién es el responsable de un trabajo de Operaciones por lotes. La presencia de etiquetas de trabajo puede conceder o limitar la capacidad de un usuario para cancelar un trabajo, activar un trabajo en estado de confirmación o cambiar el nivel de prioridad de un trabajo. Puede crear trabajos con etiquetas asociadas a ellos y agregar etiquetas a los trabajos después de que se hayan creado. Cada etiqueta es un par clave-valor que se puede incluir al crear el trabajo o se puede actualizar más tarde.

> **Warning**
>
> Las etiquetas de trabajo no deben contener información confidencial ni datos personales.

Piense en el siguiente ejemplo de etiquetado: suponga que quiere que su departamento de Finanzas cree un trabajo de Operaciones por lotes. Puede escribir una política de AWS Identity and Access Management (IAM) que permita a un usuario invocar `Finance`, dado que se ha creado el trabajo con la etiqueta `CreateJob` asignada al valor `Department`. Además, puede añadir esa política a todos los usuarios que sean miembros del departamento de Finanzas.

Siguiendo con este ejemplo, se podría escribir una política que permita a los usuarios actualizar la prioridad de cualquier trabajo que tenga las etiquetas deseadas, o cancelar cualquier trabajo con esas etiquetas. Para obtener más información, consulte the section called “Control de permisos”.

Puede añadir etiquetas a los nuevos trabajos de Operaciones por lotes de S3 cuando los cree, o puede añadírselas a trabajos ya existentes.

Tenga en cuenta las siguientes restricciones de las etiquetas:

- Puede asociar hasta 50 etiquetas a un trabajo siempre que tengan claves de etiqueta únicas.
- Una clave de etiqueta puede tener una longitud de hasta 128 caracteres Unicode y los valores de etiqueta pueden tener una longitud de hasta 256 caracteres Unicode.
- La clave y los valores distinguen entre mayúsculas y minúsculas.

Para obtener más información sobre las restricciones de las etiquetas, consulte Restricciones de las etiquetas definidas por el usuario en la Guía del usuario de AWS Billing and Cost Management.

Operaciones de API relacionadas con el etiquetado de trabajos de Operaciones por lotes de S3

Amazon S3 admite las siguientes operaciones de API específicas al etiquetado de trabajos de Operaciones por lotes de S3:

- GetJobTagging: devuelve el conjunto de etiquetas asociado a un trabajo de Operaciones por lotes.
- PutJobTagging: reemplaza el conjunto de etiquetas asociado a un trabajo. Hay dos situaciones diferentes en la administración de etiquetas de trabajos de Operaciones por lotes de S3 con esta acción de la API:
  - El trabajo no tiene etiquetas: puede añadir un conjunto de etiquetas a un trabajo (el trabajo no tiene etiquetas anteriores).
  - El trabajo tiene un conjunto de etiquetas existente: para modificar el conjunto de etiquetas existente, puede reemplazarlo todo o realizar cambios en él recuperándolo mediante GetJobTagging, modificar el conjunto de etiquetas y usar esta acción de la API para sustituir el conjunto de etiquetas con el que ha modificado.

    > **Note**
    >
    > Si envía esta solicitud con un conjunto de etiquetas vacío, Operaciones por lotes de S3 elimina el conjunto de etiquetas existente en el objeto. Si utiliza este método, se le cobrará por una solicitud de nivel 1 (PUT). Para obtener más información, consulte Precios de Amazon S3.
    >
    > Para eliminar etiquetas existentes para su trabajo de Operaciones por lotes, se prefiere la acción `DeleteJobTagging`, ya que logra el mismo resultado sin incurrir en cargos.

- DeleteJobTagging: elimina el conjunto de etiquetas asociado a un trabajo de Operaciones por lotes.

Creación de un trabajo de Operaciones por lotes con etiquetas de trabajo como etiquetado

Para etiquetar y controlar el acceso a los trabajos de Operaciones por lotes de S3, puede añadir etiquetas. Las etiquetas se pueden utilizar para identificar quién es el responsable de un trabajo de

Operaciones por lotes. Puede crear trabajos con etiquetas asociadas a ellos y agregar etiquetas a los trabajos después de que se hayan creado. Para obtener más información, consulte the section called “Uso de etiquetas”.

Mediante AWS CLI

En el siguiente ejemplo de la AWS CLI, se crea un trabajo de `S3PutObjectCopy` de la herramienta de operaciones por lotes de S3 mediante el uso de etiquetas de trabajo como marca de trabajo.

1. Seleccione la acción o el valor `OPERATION` que quiera que realice el trabajo de Operaciones por lotes y seleccione el valor `TargetResource`.

```
read -d '' OPERATION <<EOF
{
  "S3PutObjectCopy": {
    "TargetResource": "arn:aws:s3:::destination-bucket"
  }
}
EOF
```

2. Identifique la tarea `TAGS` que quiera para el trabajo. En este caso, se aplican dos etiquetas, `department` y `FiscalYear`, con los valores `Marketing` y `2020` respectivamente.

```
read -d '' TAGS <<EOF
[
  {
    "Key": "department",
    "Value": "Marketing"
  },
  {
    "Key": "FiscalYear",
    "Value": "2020"
  }
]
EOF
```

3. Especifique el valor `MANIFEST` para el trabajo de Operaciones por lotes.

```
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. Configure el valor REPORT para el trabajo de Operaciones por lotes.

```
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. Ejecute la acción `create-job` para crear su trabajo de Operaciones por lotes con los datos de entrada establecidos en los pasos previos.

```
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";
```

Uso de AWS SDK para Java

Example

En el siguiente ejemplo, se crea un trabajo de la herramienta de operaciones por lotes de S3 con etiquetas mediante 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();
}
```

Eliminación de las etiquetas de un trabajo de Operaciones por lotes de S3

Puede utilizar estos ejemplos para eliminar las etiquetas de un trabajo de Operaciones por lotes.

Mediante AWS CLI

En el siguiente ejemplo, se eliminan las etiquetas de un trabajo de la herramienta de operaciones por lotes mediante la AWS CLI.

```
aws \
    s3control delete-job-tagging \
    --account-id 123456789012 \
    --job-id Example-e25a-4ed2-8bee-7f8ed7fc2f1c \
    --region us-east-1;
```

Eliminación de las etiquetas de trabajo de un trabajo de Operaciones por lotes

Example

En el siguiente ejemplo, se eliminan las etiquetas de un trabajo de la herramienta de operaciones por lotes de S3 mediante 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);
}
```

Colocación de etiquetas de trabajo para un trabajo de Operaciones por lotes de S3 existente

Puede utilizar PutJobTagging para agregar etiquetas de trabajo a los trabajos de operaciones por lotes de S3 existentes. Para obtener más información, consulte los ejemplos siguientes.

Mediante AWS CLI

A continuación se muestra un ejemplo de cómo utilizar `s3control put-job-tagging` para agregar etiquetas de trabajo a un trabajo de la herramienta de operaciones por lotes de S3 mediante la AWS CLI.

> **Note**
>
> Si envía esta solicitud con un conjunto de etiquetas vacío, Operaciones por lotes de S3 elimina el conjunto de etiquetas existente en el objeto. Además, si utiliza este método, se le cobra por una solicitud de nivel 1 (PUT). Para obtener más información, consulte Precios de Amazon S3.
>
> Para eliminar etiquetas existentes para su trabajo de Operaciones por lotes, se prefiere la acción `DeleteJobTagging`, ya que logra el mismo resultado sin incurrir en cargos.

1. Identifique la tarea `TAGS` que quiera para el trabajo. En este caso, se aplican dos etiquetas, `department` y `FiscalYear`, con los valores `Marketing` y `2020` respectivamente.

   ```
   read -d '' TAGS <<EOF
   [
     {
       "Key": "department",
       "Value": "Marketing"
   ```

```
  },
  {
    "Key": "FiscalYear",
    "Value": "2020"
  }
]
EOF
```

2. Ejecute la acción `put-job-tagging` con los parámetros requeridos.

```
aws \
    s3control put-job-tagging \
    --account-id 123456789012 \
    --tags "${TAGS//$'\n'/}" \
    --job-id Example-e25a-4ed2-8bee-7f8ed7fc2f1c \
    --region us-east-1;
```

Uso de AWS SDK para Java

Example

En el siguiente ejemplo, se colocan las etiquetas de un trabajo de la herramienta de operaciones por lotes de S3 mediante 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);
}
```

Obtención de las etiquetas de un trabajo de Operaciones por lotes de S3

Puede utilizar `GetJobTagging` para devolver las etiquetas de un trabajo de operaciones por lotes de S3. Para obtener más información, consulte los ejemplos siguientes.

Mediante AWS CLI

En el siguiente ejemplo, se obtienen las etiquetas de un trabajo de la herramienta de operaciones por lotes mediante la AWS CLI.

```
aws \
    s3control get-job-tagging \
    --account-id 123456789012 \
    --job-id Example-e25a-4ed2-8bee-7f8ed7fc2f1c \
    --region us-east-1;
```

Uso de AWS SDK para Java

Example

En el siguiente ejemplo, se obtienen las etiquetas de un trabajo de la herramienta de operaciones por lotes de S3 mediante 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;
}
```

Control de permisos para Operaciones por lotes de S3 mediante etiquetas de trabajo

Para ayudarle a administrar sus trabajos de Operaciones por lotes de S3, puede agregar etiquetas de trabajo. Las etiquetas de trabajo le permiten controlar el acceso a sus trabajos de Operaciones por lotes y hacer que las etiquetas se apliquen cuando se cree un trabajo.

Puede aplicar hasta 50 etiquetas de trabajo a cada trabajo de Operaciones por lotes. Esto le permite establecer políticas muy pormenorizadas que restringen el conjunto de usuarios que pueden editar el trabajo. Las etiquetas de trabajo pueden conceder o limitar la capacidad de un usuario para cancelar un trabajo, activar un trabajo en estado de confirmación o cambiar el nivel de prioridad de un trabajo.

Además, puede exigir que las etiquetas se apliquen a todos los trabajos nuevos y especificar los pares clave-valor permitidos para las etiquetas. Puede expresar todas estas condiciones utilizando el mismo lenguaje de política de IAM. Para obtener más información, consulte Actions, resources, and condition keys for Amazon S3 en la Referencia de autorización de servicios.

En el ejemplo siguiente se muestra cómo puede utilizar etiquetas de trabajo de Operaciones por lotes de S3 para conceder a los usuarios permiso para crear y editar solo los trabajos que se ejecutan en un departamento específico (por ejemplo: el departamento Finanzas o Conformidad). También puede asignar trabajos en función de la etapa de desarrollo con la que están relacionados, como Control de calidad o Producción.

En este ejemplo, se utilizan etiquetas de trabajo de la herramienta de operaciones por lotes de S3 en las políticas de AWS Identity and Access Management (IAM) con el fin de conceder a los usuarios permiso para crear y editar solo los trabajos que se ejecutan en su departamento. Los trabajos se asignan en función de la etapa de desarrollo con la que están relacionados, como Control de calidad o Producción.

En este ejemplo se utilizan los siguientes departamentos, cada uno de los cuales usa las Operaciones por lotes de distinta forma:

- Finanzas
- Conformidad
- Inteligencia de negocio
- Diseño

Temas

- Control del acceso mediante la asignación de etiquetas a usuarios y recursos
- Etiquetado de trabajos de Operaciones por lotes por etapa y aplicación de límites a la prioridad del trabajo

Control del acceso mediante la asignación de etiquetas a usuarios y recursos

En este escenario, los administradores están utilizando el control de acceso basado en atributos (ABAC). ABAC es una estrategia de autorización de IAM que define los permisos adjuntando etiquetas a los usuarios y a los recursos de AWS.

A los usuarios y los trabajos se les asigna una de las siguientes etiquetas de departamento:

Clave : valor

- `department : Finance`
- `department : Compliance`
- `department : BusinessIntelligence`
- `department : Engineering`

> **Note**
>
> Las claves y los valores de las etiquetas de trabajo distinguen entre mayúsculas y minúsculas.

Mediante la estrategia de control de acceso de ABAC, se concede a un usuario del departamento de Finanzas permiso para crear y administrar trabajos de Operaciones por lotes de S3 en el departamento al asociar la etiqueta `department=Finance` con el usuario.

Además, puede adjuntar una política administrada al usuario de IAM que permita a cualquier usuario de su empresa crear o modificar trabajos de Operaciones por lotes de S3 dentro de sus respectivos departamentos.

La política de este ejemplo incluye tres instrucciones de políticas:

- La primera instrucción de la política permite al usuario crear un trabajo de Operaciones por lotes siempre que la solicitud de creación de trabajo incluya una etiqueta de trabajo que coincida con su departamento respectivo. Esto se expresa utilizando la sintaxis `"${aws:PrincipalTag/department}"`, que se reemplaza por la etiqueta de departamento del usuario en el momento de la evaluación de políticas. La condición se cumple cuando el valor proporcionado para la etiqueta de departamento en la solicitud (`"aws:RequestTag/department"`) coincide con el departamento del usuario.
- La segunda instrucción de la política permite a los usuarios cambiar la prioridad de los trabajos o actualizar el estado de un trabajo siempre que el trabajo que está actualizando coincida con el departamento del usuario.
- La tercera instrucción permite a un usuario actualizar las etiquetas de un trabajo de Operaciones por lotes en cualquier momento a través de una solicitud `PutJobTagging`, siempre y cuando (1) se conserve su etiqueta de departamento y (2) el trabajo que esté actualizando se incluya en su departamento.

```
{
      "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}"
                }
            }
        }
    ]
}
```

Etiquetado de trabajos de Operaciones por lotes por etapa y aplicación de límites a la prioridad del trabajo

Todos los trabajos de Operaciones por lotes de S3 tienen una prioridad numérica, que Amazon S3 utiliza para decidir en qué orden ejecutar los trabajos. En este ejemplo, se restringe la prioridad máxima que la mayoría de los usuarios pueden asignar a los trabajos, con rangos de prioridad más altos reservados para un conjunto limitado de usuarios privilegiados, de la siguiente manera:

- Rango de prioridad de fase de control de calidad (bajo): 1-100
- Rango de prioridad de la etapa de producción (alto): 1-300

Para ello, introduzca un nuevo conjunto de etiquetas que represente la etapa del trabajo:

Clave : valor

- `stage : QA`
- `stage : Production`

Creación y actualización de trabajos de baja prioridad dentro de un departamento

Esta política introduce dos nuevas restricciones a la creación y actualización de trabajos de Operaciones por lotes de S3, además de la restricción basada en departamentos:

- Permite a los usuarios crear o actualizar trabajos en su departamento con una nueva condición que requiere que el trabajo incluya la etiqueta `stage=QA`.
- Permite a los usuarios crear o actualizar la prioridad de un trabajo hasta una nueva prioridad máxima 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": "*"
    }
    ]
}
```

Creación y actualización de trabajos de alta prioridad dentro de un departamento

Es posible que un pequeño número de usuarios requiera la capacidad de crear trabajos de alta prioridad en Control de calidad o Producción. Para dar soporte a esta necesidad, cree una política administrada que se adapte a la política de baja prioridad de la sección anterior.

Esta política hace lo siguiente:

- Permite a los usuarios crear o actualizar trabajos en su departamento con las etiquetas `stage=QA` o `stage=Production`.
- Permite a los usuarios crear o actualizar la prioridad de un trabajo hasta un máximo 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"
                    ]
                }
            }
        }
    ]
}
```

Administración del Bloqueo de objetos en S3 mediante Operaciones por lotes de S3

Con el bloqueo de objetos de S3 puede colocar una retención legal en una versión de objeto. Al igual que un periodo de retención, la retención legal impide que se sobrescriba o elimine una versión de un objeto. Sin embargo, una retención legal no tiene asociado un periodo de retención y sigue vigente hasta que se elimine. Para obtener más información, consulte Bloqueo de objetos de retención legal en S3.

Para obtener información acerca del uso de las operaciones por lotes de S3 con el bloqueo de objetos para agregar retenciones legales a muchos objetos de Amazon S3 a la vez, consulte las siguientes secciones.

Temas

- Habilitación de Bloqueo de objetos en S3 mediante Operaciones por lotes de S3
- Configuración de la retención de Bloqueo de objetos mediante Operaciones por lotes
- Uso de operaciones por lotes de S3 con el modo de conformidad de la retención de bloqueo de objetos de S3

- Uso de Operaciones por lotes de S3 con el modo de control de la retención de Bloqueo de objetos en S3
- Uso de operaciones por lotes de S3 para desactivar la retención legal del bloqueo de objetos de S3

Habilitación de Bloqueo de objetos en S3 mediante Operaciones por lotes de S3

Puede utilizar Operaciones por lotes de S3 con bloqueo de objetos de S3 para administrar la retención o activar una retención legal para varios objetos de Amazon S3 a la vez. Especifique la lista de objetos de destino en el manifiesto y envíela a Operaciones por lotes para su finalización. Para obtener más información, consulte the section called “Retención de bloqueo de objetos” y the section called “Bloqueo de objetos de retención legal”.

En los siguientes ejemplos, se muestra cómo crear un rol de IAM con permisos de Operaciones por lotes de S3 y actualizar los permisos del rol para crear trabajos que habilitan el Bloqueo de objetos. En los ejemplos, reemplace cualquier valor variable por aquellos que se adapten a sus necesidades. También debe tener un manifiesto CSV que identifique los objetos para su trabajo de Operaciones por lotes de S3. Para obtener más información, consulte the section called “Especificar un manifiesto”.

Mediante AWS CLI

1. Cree un rol de IAM y asigne permisos de Operaciones por lotes de S3 para su ejecución.

   Este paso es necesario para todos los trabajos de Operaciones por lotes de 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. Configure Operaciones por lotes de S3 con Bloqueo de objetos en S3 para su ejecución.

   En este paso, permite que el rol haga lo siguiente:

   a. Ejecute el bloqueo de objetos en el bucket de S3 que contenga los objetos de destino en los que desee ejecutar Operaciones por lotes.

   b. Lea el bucket de S3 donde se encuentran el archivo CSV de manifiesto y los objetos.

   c. Escriba los resultados del trabajo de Operaciones por lotes de S3 en el bucket de informes.

```
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}"
```

Uso de AWS SDK para Java

En los siguientes ejemplos, se muestra cómo crear un rol de IAM con permisos de la herramienta de operaciones por lotes de S3 y actualizar los permisos de rol para crear trabajos que habilitan el bloqueo de objetos mediante AWS SDK for Java. En el código, reemplace cualquier valor de variable por aquellos que se adapten a sus necesidades. También debe tener un manifiesto CSV que identifique los objetos para su trabajo de Operaciones por lotes de S3. Para obtener más información, consulte the section called “Especificar un manifiesto”.

Debe realizar los pasos siguientes:

1. Cree un rol de IAM y asigne permisos de Operaciones por lotes de S3 para su ejecución. Este paso es necesario para todos los trabajos de Operaciones por lotes de S3.
2. Configure Operaciones por lotes de S3 con Bloqueo de objetos en S3 para su ejecución.

   Permite que el rol realice lo siguiente:

   1. Ejecute el bloqueo de objetos en el bucket de S3 que contenga los objetos de destino en los que desee ejecutar Operaciones por lotes.
   2. Lea el bucket de S3 donde se encuentran el archivo CSV de manifiesto y los objetos.
   3. Escriba los resultados del trabajo de Operaciones por lotes de S3 en el bucket de informes.

```
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);
}
```

Configuración de la retención de Bloqueo de objetos mediante Operaciones por lotes

En el siguiente ejemplo se permite que la regla establezca la retención de Bloqueo de objetos en S3 para los objetos en el bucket del manifiesto.

Se actualiza el rol para incluir permisos `s3:PutObjectRetention` de modo que pueda ejecutar la retención de Bloqueo de objetos en los objetos del bucket.

Mediante 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}"
```

Uso de AWS SDK para 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);
}
```

Uso de operaciones por lotes de S3 con el modo de conformidad de la retención de bloqueo de objetos de S3

El siguiente ejemplo se basa en los ejemplos anteriores de creación de una política de confianza y en establecer permisos de configuración de Operaciones por lotes de S3 y Bloqueo de objetos en S3 en sus objetos. En este ejemplo se establece el modo de retención en `COMPLIANCE` y la `retain until date` del 1 de enero de 2025. Crea un trabajo que apunta a objetos del bucket del manifiesto e informa de los resultados en el bucket de informes que ha identificado.

Mediante AWS CLI

Example Establecer el cumplimiento de mención en varios objetos

```
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 Ampliar la **`COMPLIANCE`** del modo de **`retain until date`** hasta el 15 de enero de 2025

El siguiente ejemplo extiende la `COMPLIANCE` del modo de `retain until date` hasta el 15 de enero de 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";
```

Uso de AWS SDK para Java

Example Establezca el modo de retención en CUMPLIMIENTO y la fecha de retención hasta el 1 de enero de 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 Extensión de la **`COMPLIANCE`** del modo de **`retain until date`**

El siguiente ejemplo extiende la `COMPLIANCE` del modo de `retain until date` hasta el 15 de enero de 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();
}
```

Uso de Operaciones por lotes de S3 con el modo de control de la retención de Bloqueo de objetos en S3

El siguiente ejemplo se basa en el ejemplo anterior de creación de una política de confianza y en establecer permisos de configuración de Operaciones por lotes de S3 y Bloqueo de objetos en S3. Muestra cómo aplicar el control de la retención de Bloqueo de objetos en S3 con la `retain until`

`date` del 30 de enero de 2025 en varios objetos. Crea un trabajo de Operaciones por lotes que utiliza el bucket del manifiesto e informa de los resultados en el bucket de informes.

Uso de la AWS CLI

Example Aplicar el control de retención de Bloqueo de objetos en S3 en varios objetos con fecha de retención hasta el 30 de enero de 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 Omitir el control de retención en varios objetos

El siguiente ejemplo se basa en el ejemplo anterior de creación de una política de confianza y en establecer permisos de configuración de Operaciones por lotes de S3 y Bloqueo de objetos en S3. Muestra cómo omitir el control de retención en varios objetos y crea un trabajo de Operaciones por lotes que utiliza el bucket del manifiesto e informa de los resultados en el bucket de informes.

```
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";
```

Uso de AWS SDK para Java

El siguiente ejemplo se basa en el ejemplo anterior de creación de una política de confianza y en establecer permisos de configuración de Operaciones por lotes de S3 y Bloqueo de objetos en S3. Muestra cómo aplicar el control de la retención de bloqueo de objetos de S3 con `retain until date` establecido en el 30 de enero de 2025 en varios objetos. Crea un trabajo de Operaciones por lotes que utiliza el bucket del manifiesto e informa de los resultados en el bucket de informes.

Example Aplicar el control de retención de Bloqueo de objetos en S3 en varios objetos con fecha de retención hasta el 30 de enero de 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 Omitir el control de retención en varios objetos

El siguiente ejemplo se basa en el ejemplo anterior de creación de una política de confianza y en establecer permisos de configuración de Operaciones por lotes de S3 y Bloqueo de objetos en S3.

Muestra cómo omitir el control de retención en varios objetos y crea un trabajo de Operaciones por lotes que utiliza el bucket del manifiesto e informa de los resultados en el bucket de informes.

```
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();
}
```

Uso de operaciones por lotes de S3 para desactivar la retención legal del bloqueo de objetos de S3

El siguiente ejemplo se basa en los ejemplos anteriores de creación de una política de confianza y en establecer permisos de configuración de Operaciones por lotes de S3 y Bloqueo de objetos en S3. Muestra cómo deshabilitar la retención legal de Bloqueo de objetos en objetos mediante Operaciones por lotes.

En el ejemplo se actualiza primero el rol para conceder permisos `s3:PutObjectLegalHold`, se crea un trabajo de Operaciones por lotes que desactiva (quita) la retención legal de los objetos identificados en el manifiesto e informa al respecto.

Mediante AWS CLI

Example Actualiza el rol para conceder permisos **`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 Desactivar la retención legal

El siguiente ejemplo desactiva la retención legal.

```
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";
```

Uso de AWS SDK para Java

Example Actualiza el rol para conceder permisos **`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 Desactivar la retención legal

Utilice el ejemplo siguiente si desea desactivar la retención legal.

```
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();
}
```

Tutorial operaciones por lotes de S3

En el siguiente tutorial se presentan procedimientos integrales completos para algunas tareas de operaciones por lotes.

- Tutorial: videos de transcodificación por lotes con operaciones por lotes de S3, AWS Lambda, y AWS Elemental MediaConvert

Monitorización de Amazon S3

El monitoreo es una parte importante a la hora de mantener la fiabilidad, la disponibilidad y el rendimiento de Amazon S3 y las soluciones de AWS. Recomendamos que recopile los datos de supervisión de todas las partes de la solución de AWS para que le resulte más sencillo depurar un error multipunto en caso de que se produzca. Antes de comenzar a supervisar Amazon S3, debe crear un plan de supervisión que incluya respuestas a las siguientes preguntas:

- ¿Cuáles son los objetivos de la supervisión?
- ¿Qué recursos va a supervisar?
- ¿Con qué frecuencia va a supervisar estos recursos?
- ¿Qué herramientas de monitoreo va a utilizar?
- ¿Quién se encargará de realizar las tareas de monitoreo?
- ¿Quién debería recibir una notificación cuando surjan problemas?

Para obtener más información sobre el registro y la monitorización en Amazon S3, consulte los siguientes temas.

> **Note**
>
> Para obtener más información sobre el uso de la clase de almacenamiento Amazon S3 Express One Zone con buckets de directorio, consulte ¿Qué es S3 Express One Zone? y Buckets de directorio.

Temas

Herramientas de monitoreo

AWS proporciona varias herramientas que puede utilizar para monitorear Amazon S3. Puede configurar algunas de estas herramientas para que monitoricen por usted, pero otras herramientas requieren intervención manual. Le recomendamos que automatice las tareas de monitorización en la medida de lo posible.

Herramientas de monitoreo automatizadas

Puede utilizar las siguientes herramientas de monitoreo automatizadas para vigilar Amazon S3 e informar cuando haya algún problema:

- Alarmas de Amazon CloudWatch: vigile una métrica durante un periodo de tiempo especificado y realice una o varias acciones según el valor que tenga la métrica en comparación con un determinado umbral durante una serie de periodos de tiempo. La acción es una notificación enviada a un tema de Amazon Simple Notification Service (Amazon SNS) o a una política de Amazon EC2 Auto Scaling. Las alarmas de CloudWatch no invocan acciones simplemente porque se encuentren en un estado determinado. El estado debe haber cambiado y debe mantenerse durante el número de periodos especificado. Para obtener más información, consulte Monitorización de métricas con Amazon CloudWatch.
- Monitorización de registros de AWS CloudTrail: comparta archivos de registro entre cuentas, monitorice los archivos de registro de CloudTrail en tiempo real enviándolos a CloudWatch Logs, escriba aplicaciones de procesamiento de registros en Java y compruebe que los archivos de registro no hayan cambiado después de que CloudTrail los entregara. Para obtener más información, consulte Registro de llamadas a la API de Amazon S3 mediante AWS CloudTrail.

Herramientas de monitoreo manuales

Otra parte importante del monitoreo de Amazon S3 implica la monitorización manual de los elementos que no cubren las alarmas de CloudWatch. Amazon S3, CloudWatch, Trusted Advisor y otros paneles de AWS Management Console proporcionan una vista rápida del estado de su entorno de AWS. Es posible que desee habilitar el registro de acceso al servidor, que realiza un seguimiento de las solicitudes de acceso al bucket. Cada entrada del registro de acceso contiene detalles de la solicitud de acceso tales como el solicitante, el nombre del bucket, la hora de la solicitud, la acción solicitada, el estado de la respuesta y el código de error, si hay alguno. Para obtener más información, consulte Registro de solicitudes con registro de acceso al servidor.

- El panel de Amazon S3 muestra lo siguiente:
  - Los buckets, los objetos y las propiedades que contienen
- La página principal de CloudWatch muestra lo siguiente:
  - Alarmas y estado actual
  - Gráficos de alarmas y recursos
  - Estado de los servicios

  Además, puede utilizar CloudWatch para hacer lo siguiente:
  - Crear paneles personalizados para monitorear los servicios que le interesan.
  - Realizar un gráfico con los datos de las métricas para resolver problemas y descubrir tendencias.
  - Buscar y examinar todas sus métricas de recursos de AWS.
  - Crear y editar las alarmas de notificación de problemas.
- AWS Trusted Advisor puede ayudarlo a monitorear los recursos de AWS para mejorar el rendimiento, la fiabilidad, la seguridad y la rentabilidad. Hay cuatro comprobaciones de Trusted Advisor disponibles para todos los usuarios y hay más de 50 comprobaciones disponibles para usuarios con un plan de soporte Business o Enterprise. Para obtener más información, consulte AWS Trusted Advisor.

  Trusted Advisor cuenta con este tipo de comprobaciones relacionadas con Amazon S3:
  - Comprobaciones de la configuración de registro de los buckets de Amazon S3.
  - Comprobaciones de seguridad de los buckets de Amazon S3 que tienen permisos de acceso abierto.
  - Comprobaciones de la tolerancia a errores de los buckets de Amazon S3 que no tienen activado el control de versiones, o que lo tienen suspendido.

Opciones de registro para Amazon S3

Puede registrar las acciones que realizan los usuarios, los roles o los Servicios de AWS en recursos de Amazon S3 y mantener registros para fines de auditoría y conformidad. Para ello, puede utilizar el registro de acceso al servidor, el registro de AWS CloudTrail o una combinación de ambos. Le recomendamos que utilice CloudTrail para registrar acciones en el nivel de bucket y objeto para los recursos de Amazon S3. Para obtener más información acerca de cada opción, consulte las secciones siguientes:

- Registro de solicitudes con registro de acceso al servidor
- Registro de llamadas a la API de Amazon S3 mediante AWS CloudTrail

En la tabla siguiente, se enumeran las propiedades clave de los registros de CloudTrail y los registros de acceso al servidor de Amazon S3. Para asegurarse de que CloudTrail cumple con los requisitos de seguridad, revise la tabla y las notas.

Propiedades del registro	AWS CloudTrail	Registros del servidor de Amazon S3
Se pueden reenviar a otros sistemas (Registros de Amazon CloudWatch, Eventos de Amazon CloudWatch)	Sí	No
Enviar los registros a varios destinos (por ejemplo: enviar los mismos registros a dos buckets diferentes)	Sí	No
Habilitar los registros para un subconjunto de objetos (prefijo)	Sí	No
Envío de registros entre cuentas (bucket de origen y destino propiedad de cuentas diferentes)	Sí	No
Validación de la integridad del archivo de registro mediante el uso de la firma digital o el algoritmo hash	Sí	No
Selección del cifrado o uso del cifrado predeterminado para los archivos de registro	Sí	No

Propiedades del registro	AWS CloudTrail	Registros del servidor de Amazon S3
Operaciones con objetos (mediante las API de Amazon S3)	Sí	Sí
Operaciones con buckets (mediante las API de Amazon S3)	Sí	Sí
Búsqueda de registros en la interfaz de usuario	Sí	No
Campos para parámetros de bloqueo de objetos, propiedades seleccionadas de Amazon S3 para el historial de registro	Sí	No
Campos para `Object Size`, `Total Time`, `Turn-Around Time` y `HTTP Referer` para los registros	No	Sí
Transiciones, finalizaciones, restauraciones del ciclo de vida	No	Sí
Registro de claves en una operación de eliminación por lotes	No	Sí
Errores de autenticación[1]	No	Sí
Cuentas a las que se envían los registros	Propietario del bucket[2] y solicitante	Solo propietario del bucket
Performance and Cost	AWS CloudTrail	Amazon S3 Server Logs

Propiedades del registro	AWS CloudTrail	Registros del servidor de Amazon S3
Precio	Los eventos de administración (primera entrega) son gratuitos; se aplica un cargo a los eventos de datos, además del cargo de almacenamiento de registros	No hay ningún cargo adicional aparte del cargo por almacenar los registros
Velocidad de entrega de registros	Eventos de datos cada 5 minutos; eventos de administración cada 15 minutos	Unas horas
Formato de registro	JSON	Archivo de registro con entradas separadas por espacios y delimitadas por una nueva línea

Notas

1. CloudTrail no envía registros para las solicitudes que no superan la autenticación (en las que las credenciales proporcionadas no son válidas). Sin embargo, sí incluye los registros de las solicitudes que no superan el proceso de autorización (`AccessDenied`) y las solicitudes realizadas por usuarios anónimos.
2. El propietario del bucket de S3 recibe registros de CloudTrail cuando la cuenta no tiene acceso completo al objeto de la solicitud. Para obtener más información, consulte Acciones en el nivel de objeto de Amazon S3 en escenarios entre cuentas.
3. S3 no admite la entrega de registros de CloudTrail ni de registros de acceso al servidor al solicitante ni al propietario del bucket para las solicitudes de puntos de conexión de VPC cuando la política de puntos de conexión de VPC las deniega o para las solicitudes que fallan antes de que la política de VPC se evalúe.

Registro de llamadas a la API de Amazon S3 mediante AWS CloudTrail

Amazon S3 está integrado con AWS CloudTrail, un servicio que proporciona un registro de las acciones realizadas por un usuario, un rol o un Servicio de AWS. CloudTrail captura todas las llamadas a la API para Amazon S3 como eventos. Las llamadas capturadas incluyen las llamadas desde la consola de Amazon S3 y las llamadas de código hacia las operaciones de la API de Amazon S3. Mediante la información recopilada por CloudTrail, puede determinar la solicitud que se realizó a Amazon S3, la dirección IP desde la que se realizó, cuándo se realizó y detalles adicionales.

Cada entrada de registro o evento contiene información sobre quién generó la solicitud. La información de identidad del usuario lo ayuda a determinar lo siguiente:

- Si la solicitud se realizó con las credenciales del usuario raíz o del usuario.
- Si la solicitud se realizó en nombre de un usuario de IAM Identity Center.
- Si la solicitud se realizó con credenciales de seguridad temporales de un rol o fue un usuario federado.
- Si la solicitud la realizó otro Servicio de AWS.

CloudTrail está activado en la Cuenta de AWS cuando usted crea la cuenta y tiene acceso automático al Historial de eventos de CloudTrail. El Historial de eventos de CloudTrail proporciona un registro visible e inmutable, que se puede buscar y descargar, de los últimos 90 días de eventos de gestión registrados en una Región de AWS. Para obtener más información, consulte Trabajar con el historial de eventos de CloudTrail en la Guía del usuario de AWS CloudTrail. No se cobran cargos de CloudTrail por ver el Historial de eventos.

Para mantener un registro permanente de los eventos en su Cuenta de AWS más allá de los 90 días, cree un registro de seguimiento o un almacén de datos de eventos de CloudTrail Lake.

Registros de seguimiento de CloudTrail

Un registro de seguimiento permite a CloudTrail enviar archivos de registro a un bucket de Amazon S3. Todos los registros de seguimiento que cree con la AWS Management Console son de varias regiones. Puede crear un registro de seguimiento de una sola región o de varias regiones mediante la AWS CLI. Se recomienda crear un registro de seguimiento de varias regiones, ya que registra actividad en todas las Regiones de AWS de su cuenta. Si crea un

registro de seguimiento de una sola región, solo podrá ver los eventos registrados en la Región de AWS del registro de seguimiento. Para obtener más información acerca de los registros de seguimiento, consulte Creación de un registro de seguimiento para su Cuenta de AWS y Creación de un registro de seguimiento para una organización en la Guía del usuario de AWS CloudTrail.

Puede crear un registro de seguimiento para enviar una copia de los eventos de administración en curso en su bucket de Amazon S3 sin costo alguno desde CloudTrail; sin embargo, hay cargos por almacenamiento en Amazon S3. Para obtener más información sobre los precios de CloudTrail, consulte Precios de AWS CloudTrail. Para obtener información acerca de los precios de Amazon S3, consulte Precios de Amazon S3.

Almacenes de datos de eventos de CloudTrail Lake

CloudTrail Lake le permite ejecutar consultas basadas en SQL sobre los eventos. CloudTrail Lake convierte los eventos existentes en formato JSON basado en filas al formato ORC de Apache. ORC es un formato de almacenamiento en columnas optimizado para una recuperación rápida de datos. Los eventos se agregan en almacenes de datos de eventos, que son recopilaciones inmutables de eventos en función de criterios que se seleccionan aplicando selectores de eventos avanzados. Los selectores que se aplican a un almacén de datos de eventos controlan los eventos que perduran y están disponibles para la consulta. Para obtener más información acerca de CloudTrail Lake, consulte Trabajar con AWS CloudTrail Lake en la Guía del usuario de AWS CloudTrail.

Los almacenes de datos de eventos de CloudTrail Lake y las consultas generan costos adicionales. Cuando crea un almacén de datos de eventos, elige la opción de precios que desea utilizar para él. La opción de precios determina el costo de la incorporación y el almacenamiento de los eventos, así como el periodo de retención predeterminado y máximo del almacén de datos de eventos. Para obtener más información sobre los precios de CloudTrail, consulte Precios de AWS CloudTrail.

Puede almacenar los archivos de registro en el bucket durante el tiempo que desee, pero también puede definir reglas de ciclo de vida de Amazon S3 para archivar o eliminar archivos de registro automáticamente. De forma predeterminada, los archivos de registro se cifran mediante cifrado en el lado de servidor (SSE) de Amazon S3;.

Uso de registros de CloudTrail con los registros de acceso al servidor de Amazon S3 y CloudWatch Logs

Los registros de AWS CloudTrail contienen las acciones realizadas por un usuario, un rol o un servicio de AWS en Amazon S3, mientras que los registros de acceso al servidor de Amazon S3 contienen los detalles de las solicitudes que se realizan a un bucket de S3. Para obtener más información acerca del funcionamiento de los distintos registros y sus propiedades, rendimiento y costos, consulte the section called “Opciones de registro”.

Puede utilizar los registros de AWS CloudTrail junto con los registros de acceso al servidor para Amazon S3. Los registros de CloudTrail le proporcionan un seguimiento detallado de la API para las operaciones de bucket y de objeto de Amazon S3. Los registros de acceso al servidor de Amazon S3 le proporcionan visibilidad de las operaciones de nivel de objeto realizadas en sus datos en Amazon S3. Para obtener más información sobre los registros de acceso al servidor, consulte Registro de solicitudes con registro de acceso al servidor.

También puede utilizar registros de CloudTrail junto con Amazon CloudWatch para Amazon S3. La integración de CloudTrail con CloudWatch Logs envía la actividad de API de bucket de S3 obtenido por CloudTrail a una secuencia de registros de CloudWatch en el grupo de registros de CloudWatch que se especifique. Puede crear alarmas de CloudWatch para monitorear actividades específicas de la API y recibir notificaciones por correo electrónico cuando se producen las actividades de la API en cuestión. Para obtener más información acerca de las alarmas de CloudWatch para monitorear actividades específicas de la API, consulte la Guía del usuario de AWS CloudTrail. Para obtener más información acerca del uso de CloudWatch con Amazon S3, consulte Monitorización de métricas con Amazon CloudWatch.

> **Note**
>
> S3 no admite la entrega de registros de CloudTrail al solicitante ni al propietario del bucket para las solicitudes de puntos de conexión de VPC cuando la política de puntos de conexión de VPC las deniega.

Seguimiento de CloudTrail con llamadas a la API de SOAP de Amazon S3

CloudTrail realiza el seguimiento de las llamadas a la API de SOAP de Amazon S3 La compatibilidad con SOAP de Amazon S3 por HTTP está obsoleta, pero aún se encuentra disponible con HTTPS.

Para obtener más información sobre la compatibilidad de SOAP en Amazon S3, consulte Apéndice A: Usar la API de SOAP.

> **Important**
>
> Las características más recientes de Amazon S3 no son compatibles con SOAP. Le recomendamos que utilice la API de REST o los SDK de AWS.

Acciones SOAP de Amazon S3 seguidas por el registro de CloudTrail

Nombre de la API SOAP	Nombre de evento de API utilizado en el registro de CloudTrail
ListAllMyBuckets	`ListBuckets`
CreateBucket	`CreateBucket`
DeleteBucket	`DeleteBucket`
GetBucketAccessControlPolicy	`GetBucketAcl`
SetBucketAccessControlPolicy	`PutBucketAcl`
GetBucketLoggingStatus	`GetBucketLogging`
SetBucketLoggingStatus	`PutBucketLogging`

Para obtener más información acerca de CloudTrail y Amazon S3, consulte los siguientes temas:

Temas

- Eventos de Amazon S3 CloudTrail
- Entradas de archivos de registro de CloudTrail para Amazon S3 y S3 en Outposts
- Habilitación del registro de eventos de CloudTrail para buckets y objetos de S3
- Identificación de solicitudes de Amazon S3 mediante CloudTrail

Eventos de Amazon S3 CloudTrail

> **Important**
>
> Amazon S3 aplica ahora el cifrado del servidor con claves administradas por Amazon S3 (SSE-S3) como el nivel básico de cifrado para cada bucket de Amazon S3. Desde el 5 de enero de 2023, todas las cargas de objetos nuevos a Amazon S3 se cifran automáticamente sin costo adicional y sin afectar al rendimiento. El estado de cifrado automático para la configuración de cifrado predeterminada en el bucket de S3 y para cargas de objetos nuevos está disponible en registros de AWS CloudTrail, Inventario de S3, Lente de almacenamiento de S3, la consola de Amazon S3 y como encabezado de respuesta a la API de Amazon S3 adicional en AWS Command Line Interface y los SDK de AWS. Para obtener más información, consulte Preguntas frecuentes del cifrado predeterminado.

En esta sección, se proporciona información acerca de los eventos que S3 registra en CloudTrail.

Eventos de datos de Amazon S3 en CloudTrail

Los eventos de datos proporcionan información sobre las operaciones de recursos realizadas en o dentro de un recurso (por ejemplo, leer o escribir en un objeto de Amazon S3). Se denominan también operaciones del plano de datos. Los eventos de datos suelen ser actividades de gran volumen. De forma predeterminada, CloudTrail no registra eventos de datos. El Historial de eventos de CloudTrail no registra los eventos de datos.

Se aplican cargos adicionales a los eventos de datos. Para obtener más información sobre los precios de CloudTrail, consulte Precios de AWS CloudTrail.

Puede registrar eventos de datos para los tipos de recursos de Amazon S3 mediante la consola de CloudTrail, la AWS CLI o las operaciones de la API de CloudTrail. Para obtener más información sobre cómo registrar los eventos de datos, consulte Registro de eventos de datos con la AWS Management Console y Registro de eventos de datos con la AWS Command Line Interface en la Guía del usuario de AWS CloudTrail.

En la siguiente tabla se muestran los tipos de recursos de Amazon S3 para los que puede registrar eventos de datos. La columna Tipo de evento de datos (consola) muestra el valor que se debe elegir en la lista de tipos de eventos de datos de la consola de CloudTrail. La columna resources.type value muestra el valor de `resources.type`, que especificaría al configurar los selectores de eventos

avanzados mediante la AWS CLI o las API de CloudTrail. La columna API de datos registradas en CloudTrail muestra las llamadas a la API registradas en CloudTrail para el tipo de recurso.

Tipo de evento de datos (consola)	resources.type value	API de datos registradas en 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 • RestoreObject • SelectObjectContent

Tipo de evento de datos (consola)	resources.type value	API de datos registradas en CloudTrail
		• UploadPart • UploadPartCopy
S3 Express One Zone	`AWS::S3Express::Object`	• AbortMultipartUpload • CompleteMultipartUpload • CreateSession • CopyObject • CreateMultipartUpload • DeleteObject • DeleteObjects • GetObject • GetObjectAttributes • HeadBucket • HeadObject • ListObjectsV2 • ListParts • PutObject • UploadPart • UploadPartCopy

Tipo de evento de datos (consola)	resources.type value	API de datos registradas en CloudTrail
Punto de acceso de S3	`AWS::S3::Access Point`	• AbortMultipartUpload • CompleteMultipartUpload • CopyObject (solo copias de la misma región) • CreateMultipartUpload • DeleteObject • DeleteObjectTagging • GetBucketAcl • GetBucketCors • GetBucketLocation • GetBucketNotificationConfiguration • GetBucketPolicy • GetObject • GetObjectAcl • GetObjectAttributes • GetObjectLegalHold • GetObjectRetention • GetObjectTagging • HeadBucket • HeadObject • ListMultipartUploads • ListObjects • ListObjectsV2 • ListObjectVersions • ListParts • Presign • PutObject

Tipo de evento de datos (consola)	resources.type value	API de datos registradas en CloudTrail
		• PutObjectLegalHold • PutObjectRetention • PutObjectAcl • PutObjectTagging • RestoreObject • UploadPart • UploadPartCopy (solo copias de la misma región)

Tipo de evento de datos (consola)	resources.type value	API de datos registradas en CloudTrail
S3 Object Lambda	`AWS::S3ObjectLambda::AccessPoint`	• AbortMultipartUpload • CompleteMultipartUpload • CopyObject (solo copias de la misma región) • CreateMultipartUpload • DeleteObject • DeleteObjectTagging • GetObject • GetObjectAcl • GetObjectLegalHold • GetObjectRetention • GetObjectTagging • HeadObject • ListMultipartUploads • ListObjects • ListObjectVersions • ListParts • PutObject • PutObjectLegalHold • PutObjectRetention • PutObjectAcl • PutObjectTagging • RestoreObject • UploadPart • WriteGetObjectResponse

Tipo de evento de datos (consola)	resources.type value	API de datos registradas en CloudTrail
S3 Outposts	`AWS::S3Outposts::Object`	• AbortMultipartUpload • CompleteMultipartUpload • CopyObject (solo copias de la misma región) • CreateMultipartUpload • DeleteObject • DeleteObjectTagging • GetObject • GetObjectTagging • HeadObject • ListMultipartUploads • ListObjects • ListObjectsV2 • ListParts • PutObject • PutObjectTagging • UploadPart • UploadPartCopy

Puede configurar selectores de eventos avanzados para filtrar según los campos `eventName`, `readOnly` y `resources.ARN` y así registrar solo los eventos que son importantes para usted. Para obtener más información acerca de estos campos, consulte AdvancedFieldSelector en la Referencia de la API de AWS CloudTrail.

Eventos de administración de Amazon S3 en CloudTrail

Amazon S3 registra todas las operaciones de plano de control como eventos de administración. Para obtener más información sobre las operaciones de la API de S3, consulte la Referencia de la API de Amazon S3.

Cómo obtiene CloudTrail las solicitudes realizadas a Amazon S3

De forma predeterminada, CloudTrail registra las llamadas a la API de bucket de S3 que se realizaron en los últimos 90 días, pero no las solicitudes realizadas a objetos. Las llamadas en el nivel de bucket incluyen eventos como `CreateBucket`, `DeleteBucket`, `PutBucketLifecycle`, `PutBucketPolicy`, etc. Puede ver los eventos a nivel bucket en la consola de CloudTrail. Sin embargo, allí no puede ver los eventos de datos (llamadas de objeto de Amazon S3); para ello, debe analizar o consultar los registros de CloudTrail.

Acciones de cuenta de Amazon S3 seguidas por el registro de CloudTrail

CloudTrail registra las acciones de cuenta. Los registros de Amazon S3 se crean junto con otros registros de Servicio de AWS en un archivo de registros. CloudTrail determina cuándo debe crearse un nuevo archivo y escribir en él en función del periodo de tiempo y del tamaño del archivo.

En las tablas de esta sección, se enumeran las acciones en el nivel de acción de Amazon S3 compatibles con el registro por parte de CloudTrail.

Las acciones de la API de nivel de cuenta de Amazon S3 rastreadas por el registro de CloudTrail aparecen con los siguientes nombres de evento: Los nombres de los eventos de CloudTrail difieren del nombre de la acción de la API. Por ejemplo, DeletePublicAccessBlock es DeleteAccountPublicAccessBlock.

- DeleteAccountPublicAccessBlock
- GetAccountPublicAccessBlock
- PutAccountPublicAccessBlock

Acciones de bucket de Amazon S3 de las que realiza un seguimiento el registro de CloudTrail

De forma predeterminada, CloudTrail registra acciones en el nivel de bucket de uso general. Los registros de Amazon S3 se crean junto con otros registros de servicios de AWS en un archivo de registros. CloudTrail determina cuándo debe crearse un nuevo archivo y escribir en él en función del periodo de tiempo y del tamaño del archivo.

En esta sección se enumeran las acciones en el nivel de bucket de Amazon S3 compatibles con el registro por parte de CloudTrail.

Las acciones de la APIen el nivel del bucket de Amazon S3 rastreadas por el registro de CloudTrail aparecen como los siguientes nombres de evento. En algunos casos, el nombre del evento de CloudTrail difiere del nombre de la acción de la API. Por ejemplo, `PutBucketLifecycleConfiguration` es `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

Además de estas operaciones de la API, también puede usar la acción en el nivel de objeto OPTIONS object. Esta acción se trata como una acción de bucket en el registro de CloudTrail porque la acción comprueba la configuración de CORS de un bucket.

Acciones de Amazon S3 Express One Zone en el bucket (punto de conexión de API regional) a las que se ha realizado un seguimiento mediante el registro de CloudTrail

De forma predeterminada, CloudTrail registra acciones en el nivel de bucket para buckets de directorio como eventos de administración. `eventsource` para eventos de administración de CloudTrail para S3 Express One Zone es `s3express.amazonaws.com`.

Las siguientes operaciones de la API de puntos de conexión regionales se registran en CloudTrail.

- CreateBucket
- DeleteBucket
- DeleteBucketPolicy
- GetBucketPolicy
- PutBucketPolicy
- ListDirectoryBuckets

Para obtener más información, consulte Registro con AWS CloudTrail para S3 Express One Zone.

Acciones en el nivel de objeto de Amazon S3 en escenarios entre cuentas

Los siguientes casos son casos de uso especiales relativos a las llamadas a la API de objeto en situaciones que afectan a varias cuentas y a la manera de informar de los registros de CloudTrail. CloudTrail entrega los registros al solicitante (la cuenta que realiza la llamada a la API), excepto en algunos casos de acceso denegado en los que las entradas de registro se redactan u omiten. Al configurar el acceso a varias cuentas, tenga en cuenta los ejemplos de esta sección.

> **Note**
>
> En los ejemplos, se presupone que los registros de CloudTrail están configurados correctamente.

Ejemplo 1: CloudTrail entrega los registros al propietario del bucket

CloudTrail entrega los registros al propietario del bucket aunque este no tenga permisos para la misma operación de API de objetos. Piense en el siguiente escenario con varias cuentas:

- La cuenta A es la propietaria del bucket.
- La cuenta B (el solicitante) intenta acceder a un objeto de ese bucket.
- La cuenta C es la propietaria del objeto. Es posible o no que la cuenta C sea la misma cuenta que la cuenta A.

> **Note**
>
> CloudTrail siempre entrega los registros de la API en el nivel de objeto al solicitante (cuenta B). Además, CloudTrail también entrega los mismos registros al propietario del bucket (cuenta A) incluso cuando el propietario del bucket no tiene el objeto (cuenta C) o tiene permisos para esas mismas operaciones de la API en dicho objeto.

Ejemplo 2: CloudTrail no hace proliferar las direcciones de correo electrónico empleadas al establecer las ACL de objeto

Piense en el siguiente escenario con varias cuentas:

- La cuenta A es la propietaria del bucket.
- La cuenta B (el solicitante) envía una solicitud para establecer un permiso de la ACL de un objeto con una dirección de correo electrónico. Para obtener más información acerca de las ACL, consulte Información general de las Listas de control de acceso (ACL).

El solicitante obtiene los registros junto con la información del correo electrónico. Sin embargo, el propietario del bucket (si puede recibir registros como en el ejemplo 1) obtiene el registro de CloudTrail que informa del evento. Aun así, el propietario del bucket no obtiene la información sobre

la configuración de ACL, específicamente la dirección de correo electrónico del receptor del permiso y el permiso en sí. La única información que el registro comunica al propietario del bucket es que la cuenta B realizó una llamada a la API de ACL.

Entradas de archivos de registro de CloudTrail para Amazon S3 y S3 en Outposts

> **Important**
>
> Amazon S3 aplica ahora el cifrado del servidor con claves administradas por Amazon S3 (SSE-S3) como el nivel básico de cifrado para cada bucket de Amazon S3. Desde el 5 de enero de 2023, todas las cargas de objetos nuevos a Amazon S3 se cifran automáticamente sin costo adicional y sin afectar al rendimiento. El estado de cifrado automático para la configuración de cifrado predeterminada en el bucket de S3 y para cargas de objetos nuevos está disponible en registros de AWS CloudTrail, Inventario de S3, Lente de almacenamiento de S3, la consola de Amazon S3 y como encabezado de respuesta a la API de Amazon S3 adicional en AWS Command Line Interface y los SDK de AWS. Para obtener más información, consulte Preguntas frecuentes del cifrado predeterminado.

Un evento representa una única solicitud de cualquier origen e incluye información sobre la operación de la API solicitada, la fecha y la hora de la operación, los parámetros de la solicitud, entre otras cosas. Los archivos de registro de CloudTrail no rastrean el orden en la pila de las llamadas a la API públicas, por lo que los eventos no aparecen en un orden específico.

> **Note**
>
> Para ver ejemplos de archivos de registro de CloudTrail para Amazon S3 Express One Zone, consulte CloudTrail log file examples for S3 Express One Zone.

Para obtener más información, consulte los ejemplos siguientes.

Temas

- Ejemplo: entrada de archivo de registro de CloudTrail para Amazon S3
- Ejemplo: entradas de archivo de registro de Amazon S3 en Outposts

Ejemplo: entrada de archivo de registro de CloudTrail para Amazon S3

En el ejemplo siguiente, se muestra una entrada de registro de CloudTrail que ilustra las acciones Servicio GET, PutBucketAcl y GetBucketVersioning.


```
{
    "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": "amzn-s3-demo-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"
    }
    }
  ]
}
```

Ejemplo: entradas de archivo de registro de Amazon S3 en Outposts

Los eventos de administración de Amazon S3 en Outposts están disponibles a través de AWS CloudTrail. Para obtener más información, consulte Registro de llamadas a la API de Amazon S3 mediante AWS CloudTrail. Además, puede habilitar el registro de eventos de datos en AWS CloudTrail de forma opcional.

Un registro de seguimiento es una configuración que permite la entrega de eventos como archivos de registro al bucket de S3 en una región que especifique. Los registros de CloudTrail para sus buckets de Outposts incluyen un nuevo campo, `edgeDeviceDetails`, que identifica Outposts donde se encuentra el bucket especificado.

Los campos de registro adicionales incluyen la acción solicitada, la fecha y hora de la acción y los parámetros de solicitud. Los archivos de registro de CloudTrail no son un seguimiento de pila ordenado de las llamadas a la API públicas, por lo que no aparecen en un orden específico.

En el ejemplo siguiente se muestra una entrada de registro de CloudTrail que muestra una acción PutObject en `s3-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": "amzn-s3-demo-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"
    }
```

Habilitación del registro de eventos de CloudTrail para buckets y objetos de S3

Puede utilizar eventos de datos de CloudTrail para obtener información sobre solicitudes de bucket y objeto en Amazon S3. Para habilitar los eventos de datos de CloudTrail para todos los buckets o para una lista de buckets específicos, debe crear un registro de seguimiento manualmente en CloudTrail.

> **Note**
>
> - La configuración predeterminada de CloudTrail es encontrar solo los eventos de administración. Asegúrese de que tiene habilitados los eventos de datos en la cuenta.

- Con un bucket de S3 que genera una gran carga de trabajo, podrían generarse rápidamente miles de registros en un corto periodo de tiempo. Piense bien el tiempo que va a tener activados los eventos de datos de CloudTrail para un bucket ocupado.

CloudTrail almacena los registros de eventos de datos de Amazon S3 en el bucket de S3 que usted elija. Considere la posibilidad de utilizar un bucket de una Cuenta de AWS distinta para organizar mejor los eventos procedentes de varios buckets que es posible que tenga en un lugar central y, de ese modo, simplificar las consultas y los análisis. AWS Organizations le ayuda a crear una Cuenta de AWS que esté vinculada a la cuenta que contiene el bucket que monitorea. Para obtener más información, consulte ¿Qué es AWS Organizations? en la Guía del usuario de AWS Organizations.

Cuando registra los eventos de datos de un registro de seguimiento en CloudTrail, tiene la opción de usar selectores de eventos avanzados o selectores de eventos básicos para registrar eventos de datos para objetos almacenados en buckets de uso general.. Para registrar eventos de datos de objetos almacenados en buckets de directorio, debe utilizar selectores de eventos avanzados. Para obtener más información, consulte Registro con AWS CloudTrail para S3 Express One Zone.

Cuando cree un registro de seguimiento en la consola de CloudTrail con selectores de eventos avanzados, en la sección de eventos de datos, puede elegir Registrar todos los eventos para la Plantilla de selector de registros para registrar todos los eventos en el nivel de objeto. Cuando cree un registro de seguimiento en la consola de CloudTrail con selectores de eventos básicos, en la sección de eventos de datos, puede marcar la casilla de verificación Selección de todos los buckets de S3 de la cuenta para registrar todos los eventos de nivel de objeto.

Note

- Una práctica recomendada consiste en crear una configuración de ciclo de vida para el bucket de eventos de datos de AWS CloudTrail. Defina la configuración del ciclo de vida para eliminar periódicamente los archivos de registro tras el periodo de tiempo que considere necesario para auditarlos. Esto reduce la cantidad de datos que Athena analiza para cada consulta. Para obtener más información, consulte Configuración de un ciclo de vida en un bucket.
- Para obtener más información acerca del formato del archivo de registro, consulte Registro de llamadas a la API de Amazon S3 mediante AWS CloudTrail.

- Para obtener ejemplos de cómo consultar los registros de CloudTrail, visite la publicación Analyze Security, Compliance, and Operational Activity Using AWS CloudTrail and Amazon Athena del Blog de big data de AWS.

Habilitar el registro de objetos en un bucket mediante la consola

Puede utilizar la consola de Amazon S3 para configurar una traza de AWS CloudTrail con el fin de registrar eventos de datos para objetos de un bucket de S3. CloudTrail permite que se registren operaciones de API en el nivel de objetos de Amazon S3 como, por ejemplo, `GetObject`, `DeleteObject` y `PutObject`. Estos eventos se denominan eventos de datos.

De forma predeterminada, las trazas de CloudTrail no registran eventos de datos, pero pueden configurarse para que registren eventos de datos de los buckets de S3 que usted especifique o para que registren eventos de datos de todos los buckets de Amazon S3 incluidos en la Cuenta de AWS. Para obtener más información, consulte Registro de llamadas a la API de Amazon S3 mediante AWS CloudTrail.

CloudTrail no rellena eventos de datos en el historial de eventos de CloudTrail. Además, no todas las acciones de bucket se rellenan en el historial de eventos de CloudTrail. Para obtener más información sobre las acciones de la API de nivel de bucket de Amazon S3 de las que el registro de CloudTrail realiza un seguimiento, consulte Acciones de bucket de Amazon S3 de las que realiza un seguimiento el registro de CloudTrail. Para obtener más información acerca de cómo consultar los registros de CloudTrail, consulte el artículo del Centro de conocimientos de AWS sobre el uso de patrones de filtro de Amazon CloudWatch Logs y Amazon Athena para consultar los registros de CloudTrail.

Para configurar un registro de seguimiento para registrar eventos de datos para un bucket de S3, puede utilizar la consola de AWS CloudTrail o la consola de Amazon S3. En caso de que esté configurando una traza con el fin de registrar eventos de datos para todos los buckets de Amazon S3 en su Cuenta de AWS, es más fácil utilizar la consola de CloudTrail. Para obtener información sobre el uso de la consola de CloudTrail a fin de configurar un registro de seguimiento con el objetivo de registrar eventos de datos de S3, consulte Eventos de datos en la Guía del usuario de AWS CloudTrail.

> **Important**
>
> Se aplican cargos adicionales a los eventos de datos. Para obtener más información, consulte Precios de AWS CloudTrail.

En el siguiente procedimiento, se muestra cómo utilizar la consola de Amazon S3 a fin de configurar un registro de seguimiento de CloudTrail con el objetivo de registrar eventos de datos para un bucket de S3.

Activación del registro de eventos de datos de CloudTrail para objetos en un bucket de uso general de S3 o en un bucket de directorio de S3

1. Inicie sesión AWS Management Console Management Console y abra la consola de Amazon S3 en https://console.aws.amazon.com/s3/.
2. En la lista Buckets (Buckets), elija el nombre del bucket.
3. Seleccione Properties (Propiedades).
4. En AWS CloudTrail data events (Eventos de datos de CloudTrail), elija Configure in CloudTrail (Configurar en CloudTrail).

   Puede crear un nuevo seguimiento de CloudTrail o reutilizar uno existente y configurar eventos de datos de Amazon S3 para que se registren en el seguimiento. Para obtener información acerca de cómo crear seguimientos en la consola de CloudTrail, consulte Creación y actualización de un seguimiento con la consola en la Guía del usuario de AWS CloudTrail. Si quiere obtener información para configurar el registro de eventos de datos de Amazon S3 en la consola de CloudTrail, consulte el registro de eventos de datos para objetos de Amazon S3 en la guía del usuario de AWS CloudTrail.

   > **Note**
   >
   > Si utiliza la consola de CloudTrail o la consola de Amazon S3 para configurar un registro de seguimiento para registrar eventos de datos de registro para un bucket de S3, la consola de Amazon S3 muestra que los registros de objeto están habilitados para el bucket.

Para desactivar el registro de eventos de datos de CloudTrail para objetos en un bucket de S3

1. Inicie sesión en la AWS Management Console y abra la consola de CloudTrail en https://console.aws.amazon.com/cloudtrail/.
2. En el panel de navegación de la izquierda, elija Registros de seguimiento.
3. Elija el nombre del registro de seguimiento que ha creado para registrar los eventos de su bucket.
4. En la página de detalles del registro de seguimiento, seleccione Detener registro en la esquina superior derecha.
5. En el cuadro de diálogo que aparece, elija Detener registro.

Para obtener más información sobre la habilitación de registro de objeto al crear un bucket de S3, consulte Crear un bucket.

Para obtener más información sobre el registro de CloudTrail con buckets S3, consulte los siguientes temas:

- Visualización de las propiedades para un bucket de S3
- Registro de llamadas a la API de Amazon S3 mediante AWS CloudTrail
- Trabajo con archivos de registro de CloudTrail en la Guía del usuario de AWS CloudTrail

Identificación de solicitudes de Amazon S3 mediante CloudTrail

En Amazon S3, puede identificar solicitudes mediante un registro de eventos de AWS CloudTrail. AWS CloudTrail es la forma preferida de identificar las solicitudes de Amazon S3, pero si utiliza los registros de acceso al servidor de Amazon S3, consulte the section called “Identificación de solicitudes de S3”.

Temas

- Identificación de solicitudes realizadas a Amazon S3 en un registro de CloudTrail
- Identificación de solicitudes de firma de Amazon S3 versión 2 mediante CloudTrail
- Identificación del acceso a objetos S3 mediante CloudTrail

Identificación de solicitudes realizadas a Amazon S3 en un registro de CloudTrail

Después de configurar CloudTrail para que entregue los eventos en un bucket, debe empezar a ver cómo llegan objetos al bucket de destino en la consola de Amazon S3. Estos están formateados de la siguiente manera:

`s3://amzn-s3-demo-bucket1/AWSLogs/111122223333/CloudTrail/Region/yyyy/mm/dd`

Los eventos registrados por CloudTrail se almacenan como objetos JSON comprimidos gzipped en el bucket de S3. Para encontrar solicitudes de forma eficiente, debe utilizar un servicio como Amazon Athena para indexar y consultar los registros de CloudTrail.

Para obtener más información acerca de CloudTrail y Athena, consulte Creación de la tabla para registros de AWS CloudTrail en Athena mediante proyección de particiones en la Guía del usuario de Amazon Athena.

Identificación de solicitudes de firma de Amazon S3 versión 2 mediante CloudTrail

Puede utilizar un registro de eventos de CloudTrail a fin de identificar qué versión de firma de API se utilizó para firmar una solicitud en Amazon S3. Esta capacidad es importante porque el soporte para Signature Version 2 va a finalizar (esta característica quedará obsoleta). Cuando esto suceda, Amazon S3 dejará de aceptar solicitudes que utilicen Signature Version 2, y todas las solicitudes deberán firmarse con Signature Version 4.

Le recomendamos encarecidamente que utilice CloudTrail para determinar si alguno de sus flujos de trabajo utiliza el proceso de firma de Signature Version 2. Actualice las bibliotecas y el código para que utilicen Signature Version 4 con el fin de evitar que su negocio se vea afectado.

Para obtener más información, consulte Anuncio: AWS CloudTrail para Amazon S3 agrega nuevos campos para auditar la seguridad mejorada en AWS re:Post.

> **Note**
>
> Los eventos de CloudTrail para Amazon S3 incluyen la versión de firma en los detalles de la solicitud con el nombre clave `additionalEventData`. Para encontrar la versión de firma en solicitudes realizadas para objetos en Amazon S3, como solicitudes `GET`, `PUT` y `DELETE`, debe habilitar los eventos de datos de CloudTrail. (Esta característica está desactivada de forma predeterminada).

AWS CloudTrail es el método preferido para identificar solicitudes de Signature Version 2. Si utiliza los registros de acceso al servidor de Amazon S3, consulte Identificación de solicitudes de la versión 2 de firma mediante registros de acceso de Amazon S3.

Temas

- Ejemplos de consulta de Athena para identificar solicitudes de firma de Amazon S3 versión 2
- Partición de datos de firma de versión 2

Ejemplos de consulta de Athena para identificar solicitudes de firma de Amazon S3 versión 2

Example — Seleccionar todos los eventos de Signature Version 2 e imprimir solo **`EventTime`**, **`S3_Action`**, **`Request_Parameters`**, **`Region`**, **`SourceIP`** y **`UserAgent`**

En la siguiente consulta de Athena, sustituya *`s3_cloudtrail_events_db.cloudtrail_table`* por los detalles de Athena y aumente o elimine el límite según corresponda.

```
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 — Seleccionar todos los solicitantes que envían tráfico de firma versión 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
```

Partición de datos de firma de versión 2

Si tiene una gran cantidad de datos para consultar, puede reducir los costos y el tiempo de ejecución de Athena al crear una tabla particionada.

Para ello, cree una tabla nueva con particiones como se indica a continuación.

```
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://amzn-s3-demo-bucket1/AWSLogs/111122223333/';
```

A continuación, cree cada una de las particiones. No puede obtener resultados de fechas que no ha creado.

```
ALTER TABLE s3_cloudtrail_events_db.cloudtrail_table_partitioned ADD
    PARTITION (region= 'us-east-1', year= '2019', month= '02', day= '19') LOCATION
 's3://amzn-s3-demo-bucket1/AWSLogs/111122223333/CloudTrail/us-east-1/2019/02/19/'
    PARTITION (region= 'us-west-1', year= '2019', month= '02', day= '19') LOCATION
 's3://amzn-s3-demo-bucket1/AWSLogs/111122223333/CloudTrail/us-west-1/2019/02/19/'
    PARTITION (region= 'us-west-2', year= '2019', month= '02', day= '19') LOCATION
 's3://amzn-s3-demo-bucket1/AWSLogs/111122223333/CloudTrail/us-west-2/2019/02/19/'
    PARTITION (region= 'ap-southeast-1', year= '2019', month= '02', day= '19') LOCATION
 's3://amzn-s3-demo-bucket1/AWSLogs/111122223333/CloudTrail/ap-southeast-1/2019/02/19/'
    PARTITION (region= 'ap-southeast-2', year= '2019', month= '02', day= '19') LOCATION
 's3://amzn-s3-demo-bucket1/AWSLogs/111122223333/CloudTrail/ap-southeast-2/2019/02/19/'
    PARTITION (region= 'ap-northeast-1', year= '2019', month= '02', day= '19') LOCATION
 's3://amzn-s3-demo-bucket1/AWSLogs/111122223333/CloudTrail/ap-northeast-1/2019/02/19/'
    PARTITION (region= 'eu-west-1', year= '2019', month= '02', day= '19') LOCATION
 's3://amzn-s3-demo-bucket1/AWSLogs/111122223333/CloudTrail/eu-west-1/2019/02/19/'
    PARTITION (region= 'sa-east-1', year= '2019', month= '02', day= '19') LOCATION
 's3://amzn-s3-demo-bucket1/AWSLogs/111122223333/CloudTrail/sa-east-1/2019/02/19/';
```

A continuación, puede realizar la solicitud basándose en estas particiones y no necesita cargar el bucket completo.

```
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
```

Identificación del acceso a objetos S3 mediante CloudTrail

Puede utilizar los registros de eventos de AWS CloudTrail para identificar las solicitudes de acceso a objetos de Amazon S3 para eventos de datos, como `GetObject`, `DeleteObject` y `PutObject` y descubrir información adicional sobre esas solicitudes.

En el siguiente ejemplo, se muestra cómo obtener todas las solicitudes de objeto PUT para Amazon S3 desde el registro de eventos de AWS CloudTrail.

Temas

- Ejemplos de consulta de Athena para identificar solicitudes de acceso a objetos de Amazon S3

Ejemplos de consulta de Athena para identificar solicitudes de acceso a objetos de Amazon S3

En los ejemplos de consulta de Athena siguientes, sustituya *s3_cloudtrail_events_db.cloudtrail_table* por los detalles de Athena y modifique el intervalo de fechas según corresponda.

Example — Seleccionar todos los eventos que tengan solicitudes de acceso de objetos **PUT** e imprimir solo **EventTime**, **EventSource**, **SourceIP**, **UserAgent**, **BucketName**, **object** y **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 — Seleccionar todos los eventos que tengan solicitudes de acceso de objetos **GET** e imprimir solo **EventTime**, **EventSource**, **SourceIP**, **UserAgent**, **BucketName**, **object** y **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 — Seleccionar todos los eventos de los solicitantes anónimos en un bucket en un periodo determinado e imprima solo **EventTime**, **EventName**, **EventSource**, **SourceIP**, **UserAgent**, **BucketName**, **UserARN** y **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 — Identificar todas las solicitudes que requerían una ACL para la autorización

El siguiente ejemplo de consulta de Amazon Athena muestra cómo identificar todas las solicitudes realizadas a los buckets de S3 que requerían una lista de control de acceso (ACL) para la autorización. Si la solicitud requería una ACL para la autorización, el valor `aclRequired` en `additionalEventData` es `Yes`. Si no se requirió ninguna ACL, `aclRequired` no está presente. Puede usar esta información para migrar esos permisos de ACL a las políticas de bucket adecuadas. Una vez que haya creado estas políticas de bucket, puede desactivar las ACL de estos buckets. Para obtener más información acerca de la desactivación de las ACL, consulte Requisitos previos para desactivar las 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

- Estos ejemplos de consulta también pueden ser útiles para la monitorización de la seguridad. Puede revisar los resultados de las llamadas a las operaciones `PutObject` o `GetObject` desde solicitantes o direcciones IP inesperados o no autorizados con el fin de identificar cualquier solicitud anónima que se realice a los buckets.
- Esta consulta solo recupera información de la hora a la que se habilitó el registro.

Si utiliza los registros de acceso al servidor de Amazon S3, consulte Identificación de solicitudes de acceso a objetos mediante los registros de acceso de Amazon S3.

Registro de solicitudes con registro de acceso al servidor

El registro de acceso al servidor brinda registros detallados para las solicitudes realizadas a un bucket. Los registros de acceso al servidor resultan útiles para muchas aplicaciones. Por ejemplo, la información del registro de acceso puede ser útil en auditorías de acceso y seguridad. Esta información también puede ayudarle a conocer mejor su base de clientes y entender su factura de Amazon S3.

> **Note**
>
> Los registros de acceso del servidor no registran información relacionada con errores de redirección a regiones erróneas para las regiones lanzadas después del 20 de marzo de 2019. Los errores de redirección de región incorrecta se producen cuando se realiza una solicitud de un objeto o bucket fuera de la región en la que existe el bucket.

¿Cómo habilito la entrega de registros?

Para habilitar la entrega de registros, realice los siguientes pasos básicos. Para obtener más información, consulte Habilitación del registro de acceso al servidor de Amazon S3.

1. Indique el nombre del bucket de destino (también llamado bucket objetivo). Este bucket es donde desea que Amazon S3 guarde los registros de acceso como objetos. Tanto los buckets de origen como de destino deben estar en la misma Región de AWS y ser propiedad de la misma cuenta. El bucket de destino no debe tener una configuración de período de retención predeterminada de Bloqueo de objetos de S3. El bucket de destino tampoco debe tener activado Pago por solicitante.

   Puede enviar los registros a cualquier bucket de su propiedad que se encuentre en la misma región que el bucket de origen, incluido el propio bucket de origen. Sin embargo, para una administración de registros más sencilla, le recomendamos que guarde los registros de acceso en un bucket distinto.

   Cuando los buckets de origen y destino son el mismo, se crean registros adicionales para los registros que se escriben en el bucket, lo que crear un bucle infinito de registros. No recomendamos hacer esto porque podría dar lugar a un pequeño aumento en la facturación de almacenamiento. Además, los registros adicionales sobre registros podrían hacer que resulte más difícil encontrar el registro que busca.

Si decide guardar los registros de acceso en el bucket de origen, le recomendamos que especifique un prefijo de destino para todas las claves de objeto del registro. Al especificar un prefijo, todos los nombres de los objetos de registro comienzan con una cadena común, lo que facilita la identificación de los objetos de registro.

2. (Opcional) Asigne un prefijo de destino a todas las claves de objeto de registro de Amazon S3. El prefijo de destino le facilita la localización de los objetos de registro. Por ejemplo, si especifica el valor del prefijo `logs/`, cada objeto de registro que crea Amazon S3 comienza con el prefijo `logs/` en su clave, por ejemplo:

   ```
   logs/2013-11-01-21-32-16-E568B2907131C0C0
   ```

   Si especifica el valor del prefijo `logs`, el objeto de registro aparece de la siguiente manera:

   ```
   logs2013-11-01-21-32-16-E568B2907131C0C0
   ```

   Los prefijos también son útiles para distinguir entre los buckets de origen cuando varios buckets registran en el mismo bucket de destino.

   Este prefijo puede ser útil también cuando elimina los registros. Por ejemplo, puede establecer una regla de configuración de ciclo de vida para que Amazon S3 elimine los objetos con un prefijo específico. Para obtener más información, consulte Eliminación de archivos de registro de Amazon S3.

3. (Opcional) Establezca permisos para que otros puedan obtener acceso a los registros generados. De forma predeterminada, solo el propietario del bucket siempre tiene acceso completo a los objetos de registro. Si el bucket de destino utiliza la configuración de propietario del bucket obligatorio de S3 Object Ownership para desactivar las listas de control de acceso (ACL), no podrá conceder permisos en las concesiones de destino que utilizan ACL. Sin embargo, puede actualizar la política de bucket para el bucket de destino conceda acceso a otros. Para obtener más información, consulte Administración de identidades y accesos para Amazon S3 y Permisos para entrega de registros.

4. (Opcional) Establezca un formato de clave de objeto de registro para los archivos de registro. Dispone de dos opciones para el formato de clave de objeto de registro (también conocido como formato de clave de objeto de destino):

   - Partición no basada en fechas: este es el formato de clave del objeto de registro original. Si elige este formato, el formato de clave del archivo de registro aparece de la siguiente manera:

```
[DestinationPrefix][YYYY]-[MM]-[DD]-[hh]-[mm]-[ss]-[UniqueString]
```

Por ejemplo, si especifica `logs/` como prefijo, los objetos de registro se nombran de la siguiente manera:

```
logs/2013-11-01-21-32-16-E568B2907131C0C0
```

- Partición basada en fechas: si elige una partición basada en fechas, puede elegir la hora del evento o la hora de entrega del archivo de registro como origen de fecha que se utiliza en el formato de registro. Este formato facilita la consulta de los registros.

  Si elige la partición basada en fechas, el formato de clave del archivo de registro aparece de la siguiente manera:

  ```
  [DestinationPrefix][SourceAccountId]/[SourceRegion]/[SourceBucket]/[YYYY]/[MM]/
  [DD]/[YYYY]-[MM]-[DD]-[hh]-[mm]-[ss]-[UniqueString]
  ```

  Por ejemplo, si especifica `logs/` como prefijo de destino, los objetos de registro se nombran de la siguiente manera:

  ```
  logs/123456789012/us-west-2/DOC-EXAMPLE-SOURCE-
  BUCKET/2023/03/01/2023-03-01-21-32-16-E568B2907131C0C0
  ```

  En cuanto a la hora de entrega, la hora indicada en los nombres de los archivos de registro corresponde a la hora de entrega de los archivos de registro.

  Para la entrega de la hora, el año, el mes y el día corresponden al día en que se produjo el evento, y la hora, los minutos y los segundos se establecen en `00` en la clave. Los registros que se entregan en estos archivos de registro son solo para un día específico.

Si configura los registros mediante la AWS Command Line Interface (AWS CLI), los SDK de AWS o la API de REST de Amazon S3, utilice `TargetObjectKeyFormat` para especificar el formato de la clave del objeto de registro. Para especificar una partición no basada en fechas, utilice `SimplePrefix`. Para especificar una partición basada en datos, utilice `PartitionedPrefix`. Si usa `PartitionedPrefix`, utilice `PartitionDateSource` para especificar `EventTime` o `DeliveryTime`.

Para `SimplePrefix`, el formato de clave del archivo de registro aparece de la siguiente manera:

```
[TargetPrefix][YYYY]-[MM]-[DD]-[hh]-[mm]-[ss]-[UniqueString]
```

En el caso de `PartitionedPrefix` con la hora del evento o la hora de entrega, el formato de clave del archivo de registro es el siguiente:

```
[TargetPrefix][SourceAccountId]/[SourceRegion]/[SourceBucket]/[YYYY]/[MM]/[DD]/
[YYYY]-[MM]-[DD]-[hh]-[mm]-[ss]-[UniqueString]
```

Formato de clave de objeto de registro

Amazon S3 utiliza los siguientes formatos de clave de objeto para los objetos de registro que carga en el bucket de destino:

- Partición no basada en fechas: este es el formato de clave del objeto de registro original. Si elige este formato, el formato de clave del archivo de registro aparece de la siguiente manera:

  ```
  [DestinationPrefix][YYYY]-[MM]-[DD]-[hh]-[mm]-[ss]-[UniqueString]
  ```

- Partición basada en fechas: si elige una partición basada en fechas, puede elegir la hora del evento o la hora de entrega del archivo de registro como origen de fecha que se utiliza en el formato de registro. Este formato facilita la consulta de los registros.

  Si elige la partición basada en fechas, el formato de clave del archivo de registro aparece de la siguiente manera:

  ```
  [DestinationPrefix][SourceAccountId]/[SourceRegion]/[SourceBucket]/[YYYY]/[MM]/[DD]/
  [YYYY]-[MM]-[DD]-[hh]-[mm]-[ss]-[UniqueString]
  ```

En la clave del objeto de registro, YYYY, `MM`, `DD`, hh, `mm` y `ss` son los dígitos del año, el mes, el día, la hora, los minutos y los segundos (respectivamente). Las fechas y horas se muestran en tiempo universal coordinado (UTC).

Un archivo de registro enviado en un momento específico puede contener registros escritos en cualquier momento antes de ese momento. No hay forma de saber si se enviaron o no todas las entradas de registro para un cierto intervalo de tiempo.

El componente `UniqueString` de la clave permite impedir que se sobrescriban los archivos. No tiene ningún significado y el software de procesamiento de archivos de registro debería omitirlo.

¿Cómo se envían los registros?

Amazon S3 recopila periódicamente entradas de registro de acceso, consolida los registros en archivos de registro y luego carga los archivos de registro en su bucket de destino como objetos de registro. Si habilita los registros en varios buckets de origen que identifican el mismo bucket de destino, el bucket de destino tendrá registros de acceso para todos esos buckets de origen. No obstante, cada objeto de registro informará entradas de registro de acceso para un bucket de origen específico.

Amazon S3 utiliza una cuenta especial de entrega de registros para escribir registros de acceso al servidor. Estos escritos están sujetos a las restricciones de control de acceso habituales. Le recomendamos que actualice la política de bucket en el bucket de destino para conceder acceso a la entidad principal del servicio de registro (`logging.s3.amazonaws.com`) para la entrega de registros de acceso. También puede conceder acceso para la entrega de registros de acceso al grupo de entrega de registros de S3 a través de la lista de control de acceso (ACL) del bucket. Sin embargo, no se recomienda conceder acceso al grupo de entrega de registros de S3 mediante la ACL de su bucket.

Cuando habilita el registro de acceso al servidor y concede acceso para la entrega de registros de acceso a través de la política de bucket de destino, debe actualizar la política para permitir acceso `s3:PutObject` para la entidad principal del servicio de registro. Si utiliza la consola de Amazon S3 para habilitar los registros de acceso al servidor, la consola actualiza automáticamente la política del bucket de destino para conceder estos permisos a la entidad principal del servicio de registro. Para obtener más información acerca de cómo conceder permisos para la entrega de registros de acceso al servidor, consulte Permisos para entrega de registros.

> **Note**
>
> S3 no admite la entrega de registros de CloudTrail ni de registros de acceso al servidor al solicitante ni al propietario del bucket para las solicitudes de puntos de conexión de VPC cuando la política de puntos de conexión de VPC las deniega o para las solicitudes que fallan antes de que la política de VPC se evalúe.

Configuración de propietario del bucket obligatorio de S3 Object Ownership

Si el bucket de destino utiliza la configuración de propietario del bucket obligatorio de Object Ownership, las ACL se desactivan y ya no afectan a los permisos. Debe actualizar la política del bucket de destino para conceder acceso a la entidad principal del servicio de registro. Para obtener más información acerca de la propiedad de objetos, consulte Concesión de acceso al grupo de entrega de registros de S3 para el registro de acceso al servidor.

Envío de archivos de registro de servidor según el mejor esfuerzo

Las entradas de registro de acceso al servidor se envían según el "mejor esfuerzo", es decir, en la medida que sea posible. En la mayoría de las solicitudes de registros para un bucket debidamente configurado se envían archivos de registro. La mayoría de las entradas de registro se envían en el plazo de unas horas después de su registro, pero se pueden entregar con mayor frecuencia.

No se garantiza que los registros de servidores estén completos ni que lleguen de manera puntual. La entrada de registro de una solicitud determinada puede enviarse mucho después de que la solicitud se haya procesado realmente, y es probable no se envíe en absoluto. Es posible que incluso vea una duplicación de una entrada de registro. El objetivo de los registros de servidores es darle una idea de la naturaleza del tráfico al que se enfrenta su bucket. Aunque es poco usual perder o duplicar entradas registros, tenga en cuenta que el registro del servidor no pretende ser un recuento completo de todas las solicitudes.

Debido a la naturaleza de mejor esfuerzo del registro de servidores, sus informes de uso podrían incluir una o varias solicitudes de acceso que no aparecen en un registro de servidor enviado. Puede encontrar estos informes de uso en Informes de costes y uso en la consola de AWS Billing and Cost Management.

Los cambios del estado de los registros del bucket surten efecto con el tiempo

Los cambios del estado de registros de un bucket demoran un tiempo en implementarse efectivamente en el envío de archivos de registro. Por ejemplo, si habilita los registros para un bucket, algunas solicitudes que se realizan a la hora siguiente pueden registrarse, mientras que otras no. Supongamos que cambia el bucket de destino para registros del bucket A al bucket B. Es posible que algunos registros para la siguiente hora se sigan enviando al bucket A, mientras que otros se envíen al nuevo bucket B de destino. En todos los casos, la nueva configuración finalmente se aplica sin que usted tenga que tomar medidas adicionales.

Para obtener más información acerca de los registros y archivos de registro, consulte las siguientes secciones:

Temas

Habilitación del registro de acceso al servidor de Amazon S3

El registro de acceso al servidor proporciona el historial detallado de las solicitudes que se realizan a un bucket de Amazon S3. Los registros de acceso al servidor resultan útiles para muchas aplicaciones. Por ejemplo, la información del registro de acceso puede ser útil en auditorías de acceso y seguridad. Esta información también puede ayudarle a conocer mejor su base de clientes y entender su factura de Amazon S3.

De forma predeterminada, Amazon S3 no recopila registros de acceso al servidor. Cuando activa la actividad de registro, Amazon S3 envía los registros de acceso de un bucket de origen a un bucket de destino que usted selecciona. El bucket de destino debe estar en la misma Región de AWS y Cuenta de AWS que el bucket de origen.

Una entrada de registro de acceso incluye detalles de las solicitudes realizadas a un bucket. Esta información puede incluir el tipo de solicitud, los recursos especificados en la solicitud y la hora y la fecha en que se procesó la solicitud. Para obtener más información acerca de los conceptos básicos de los registros, consulte Registro de solicitudes con registro de acceso al servidor.

Important

- Habilitar el registro de acceso al servidor en un bucket de Amazon S3 ni incurre en ningún cargo adicional. Sin embargo, los archivos de registro que recibe del sistema acumularán los cargos usuales de almacenamiento. (Puede eliminar los registros en cualquier momento). No contemplamos los cargos de transferencia de datos por la entrega de los archivos de registro, pero sí aplicamos el cargo de la tasa normal de transferencia de datos por obtener acceso a los archivos de registro.
- El bucket de destino no debe tener habilitado el registro de acceso al servidor. Puede enviar los registros a cualquier bucket de su propiedad que se encuentre en la misma región que el bucket de origen, incluido el propio bucket de origen. Sin embargo, enviar registros al bucket de origen provocará un bucle infinito de registros, por lo que no se

recomienda. Para que la administración de registros sea más sencilla, le recomendamos que guarde los registros de acceso en un bucket distinto. Para obtener más información, consulte ¿Cómo habilito la entrega de registros?

- Los buckets de S3 con Bloqueo de objetos de S3 habilitado no se pueden utilizar como buckets de destino para los registros de acceso al servidor. Su bucket de destino no debe tener una configuración de periodo de retención predeterminada.
- El bucket de destino no debe tener activado Pago por solicitante.
- Puede usar el cifrado predeterminado del bucket en el bucket de destino solo si se usa el cifrado del lado del servidor con claves administradas por Amazon S3 (SSE-S3), que usa el Estándar de cifrado avanzado de 256 bits (AES-256). No se admite el cifrado del servidor predeterminado con claves AWS Key Management Service (AWS KMS) (SSE-KMS).

Puede habilitar o desactivar el registro de acceso al servidor mediante la consola de Amazon S3, la API de Amazon S3, AWS Command Line Interface (AWS CLI) o los SDK de AWS.

Permisos para entrega de registros

Amazon S3 utiliza una cuenta especial de entrega de registros para escribir registros de acceso al servidor. Estos escritos están sujetos a las restricciones de control de acceso habituales. Para la entrega del registro de acceso, debe conceder a la entidad principal del servicio de registro acceso (`logging.s3.amazonaws.com`) a su bucket de destino.

Para conceder permisos a Amazon S3 para la entrega de registros, puede utilizar una política de bucket o listas de control de acceso (ACL) de bucket, según la configuración de S3 Object Ownership del bucket de destino. Sin embargo, le recomendamos que utilice una política de bucket en lugar de ACL.

Configuración de propietario del bucket obligatorio de S3 Object Ownership

Si el bucket de destino utiliza la configuración de propietario del bucket obligatorio de Object Ownership, las ACL se desactivan y ya no afectan a los permisos. En ese caso, debe actualizar la política de bucket para el bucket de destino para conceder acceso a la entidad principal del servicio de registro. No puede actualizar la ACL del bucket para conceder acceso al grupo de entrega de registros de S3. Tampoco puede incluir las concesiones de destino en su configuración de PutBucketLogging.

Para obtener información sobre la migración de las ACL de bucket existentes para la entrega de registros de acceso a una política de bucket, consulte Concesión de acceso al grupo de entrega de registros de S3 para el registro de acceso al servidor. Para obtener más información acerca de la propiedad de objetos, consulte Control de la propiedad de los objetos y desactivación de las ACL del bucket. Al crear buckets nuevos, las ACL están deshabilitadas de forma predeterminada.

Concesión de acceso mediante una política de bucket

Para conceder acceso mediante la política de bucket en el bucket de destino, actualice la política de bucket para conceder el permiso `s3:PutObject` a la entidad principal del servicio de registro. Si utiliza la consola de Amazon S3 para habilitar los registros de acceso al servidor, la consola actualiza automáticamente la política del bucket en el bucket destino para conceder este permiso a la entidad principal del servicio de registro. Si habilita el registro de acceso al servidor mediante programación, debe actualizar manualmente la política del bucket de destino para conceder acceso a la entidad principal del servicio de registro.

Para ver un ejemplo de una política de bucket que concede acceso a la entidad principal del servicio de registro, consulte the section called “Concesión de permisos a la entidad principal del servicio de registro mediante una política de bucket”.

Concesión de acceso mediante ACL de bucket

También puede utilizar las ACL de bucket para conceder acceso para la entrega de registros de acceso. Agregue una entrada de concesión a la ACL del bucket que concede permisos `WRITE` y `READ_ACP` para el grupo entrega de registros de S3. Sin embargo, no se recomienda conceder acceso al grupo de entrega de registros de S3 mediante la ACL del bucket. Para obtener más información, consulte Control de la propiedad de los objetos y desactivación de las ACL del bucket. Para obtener información sobre la migración de las ACL de bucket existentes para la entrega de registros de acceso a una política de bucket, consulte Concesión de acceso al grupo de entrega de registros de S3 para el registro de acceso al servidor. Para ver un ejemplo de ACL que concede acceso a la entidad principal del servicio de registro, consulte the section called “Concesión de permisos al grupo de entrega de registros mediante una ACL de bucket”.

Concesión de permisos a la entidad principal del servicio de registro mediante una política de bucket

Esta política de bucket de ejemplo concede el permiso `s3:PutObject` a la entidad principal del servicio de registro (`logging.s3.amazonaws.com`). Para utilizar esta política de bucket, reemplace *`user input placeholders`* por su propia información. En la siguiente política, *`amzn-s3-demo-destination-bucket`* es el bucket de destino donde se entregarán los registros de acceso al servidor y *`amzn-s3-demo-source-bucket`* es el bucket de origen. *`EXAMPLE-LOGGING-`*

PREFIX es el prefijo de destino opcional que desea utilizar para sus objetos de registro. *SOURCE-ACCOUNT-ID* es la Cuenta de AWS que posee el bucket de origen.

> **Note**
>
> Si hay instrucciones Deny en su política de bucket, asegúrese de que no impidan que Amazon S3 envíe registros de acceso.

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Sid": "S3ServerAccessLogsPolicy",
            "Effect": "Allow",
            "Principal": {
                "Service": "logging.s3.amazonaws.com"
            },
            "Action": [
                "s3:PutObject"
            ],
            "Resource": "arn:aws:s3:::amzn-s3-demo-destination-bucket/EXAMPLE-LOGGING-PREFIX*",
            "Condition": {
                "ArnLike": {
                    "aws:SourceArn": "arn:aws:s3:::amzn-s3-demo-source-bucket"
                },
                "StringEquals": {
                    "aws:SourceAccount": "SOURCE-ACCOUNT-ID"
                }
            }
        }
    ]
}
```

Concesión de permisos al grupo de entrega de registros mediante una ACL de bucket

> **Note**
>
> Como práctica recomendada de seguridad, Amazon S3 desactiva las listas de control de acceso (ACL) de forma predeterminada en todos los buckets nuevos. Para obtener

más información acerca de los permisos de ACL en la consola de Amazon S3, consulte Configuración de la ACL.

Aunque no recomendamos este método, puede conceder permisos al grupo de entrega de registros mediante una ACL de bucket. Sin embargo, si el bucket de destino utiliza la configuración de propietario del bucket obligatorio de Object Ownership, no se pueden configurar ACL de bucket u objeto. Tampoco puede incluir las concesiones de destino en su configuración de PutBucketLogging. En su lugar, debe utilizar una política de bucket para conceder acceso a la entidad principal del servicio de registro (`logging.s3.amazonaws.com`). Para obtener más información, consulte Permisos para entrega de registros.

En la ACL del bucket, el grupo entrega de registros se representa con la siguiente URL:

```
http://acs.amazonaws.com/groups/s3/LogDelivery
```

Para conceder los permisos `WRITE` y `READ_ACP` (lectura de ACL), agregue las siguientes concesiones a la ACL del bucket de destino.

```
<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>
```

Para obtener ejemplos de cómo agregar concesiones de ACL mediante programación, consulte Configuración de la ACL.

Important

Al habilitar el registro de acceso al servidor de Amazon S3 mediante AWS CloudFormation en un bucket y utilizar las ACL para conceder acceso al grupo de entrega de registros de S3, también debe agregar `AccessControl": "LogDeliveryWrite"` en la plantilla de

CloudFormation. Esta acción es importante porque puede otorgar esos permisos creando solo una ACL para el bucket, pero no puede crear ACL personalizadas para los buckets en CloudFormation. Solo puede usar ACL predefinidas con CloudFormation.

Para habilitar el registro de acceso al servidor

Utilice los siguientes procedimientos para habilitar el registro de acceso al servidor mediante la consola de Amazon S3, la API de REST de Amazon S3, los SDK de AWS y AWS CLI.

Uso de la consola de S3

1. Inicie sesión en la AWS Management Console y abra la consola de Amazon S3 en https://console.aws.amazon.com/s3/.
2. En la lista Buckets (Buckets), elija el nombre del bucket para el que desea activar el registro de acceso al servidor.
3. Seleccione Properties (Propiedades).
4. En la sección Server access logging (Registro de acceso al servidor), elija Edit (Editar).
5. En Registro de acceso al servidor, seleccione Habilitar.
6. En Bucket de destino, especifique un bucket y un prefijo opcional. Si especifica un prefijo, le recomendamos incluir una barra inclinada (/) después del prefijo para facilitar la búsqueda de los registros.

   > **Note**
   >
   > Especificar un prefijo con una barra (/) facilita la localización de los objetos de registro. Por ejemplo, si especifica el valor del prefijo `logs/`, cada objeto de registro que crea Amazon S3 comienza con el prefijo `logs/` en su clave de la siguiente manera:
   >
   > ```
   > logs/2013-11-01-21-32-16-E568B2907131C0C0
   > ```
   >
   > Si especifica el valor del prefijo `logs`, el objeto de registro aparece de la siguiente manera:
   >
   > ```
   > logs2013-11-01-21-32-16-E568B2907131C0C0
   > ```

7. En Formato de clave de objeto de registro, realice una de las siguientes acciones:

- Para elegir una partición no basada en fechas, elija [DestinationPrefix][YYYY]-[MM]-[DD]-[hh]-[mm]-[ss]-[UniqueString].
- Para elegir una partición basada en fechas, elija [DestinationPrefix][SourceAccountId]/[SourceRegion]/[SourceBucket]/[YYYY]/[MM]/[DD]/[YYYY]-[MM]-[DD]-[hh]-[mm]-[ss]-[UniqueString] y luego Hora de evento de S3 u Hora de entrega de archivo de registro.

8. Elija Guardar cambios.

   Cuando se habilita el registro de acceso al servidor en un bucket, la consola habilita los registros en el bucket de origen y actualiza la política de bucket para que el bucket de destino conceda el permiso `s3:PutObject` a la entidad principal del servicio de registro (`logging.s3.amazonaws.com`). Para obtener más información acerca de esta política de bucket, consulte Concesión de permisos a la entidad principal del servicio de registro mediante una política de bucket.

   Puede ver los registros en el bucket de destino. Después de habilitar el registro de acceso al servidor, la entrega de los registros al bucket de destino puede tardar unos horas. Para obtener más información acerca de cómo y cuándo se entregan los registros, consulte ¿Cómo se envían los registros?.

Para obtener más información, consulte Visualización de las propiedades para un bucket de S3.

Uso de la API de REST

Para habilitar los registros, envíe una solicitud PutBucketLogging para agregar la configuración de registros en el bucket de origen. La solicitud especifica el bucket de destino y, de forma opcional, el prefijo que se debe utilizar con todas las claves de objeto de registro.

En el siguiente ejemplo se identifica *amzn-s3-demo-destination-bucket* como el bucket de destino y *logs/* como el prefijo.

```
<BucketLoggingStatus xmlns="http://doc.s3.amazonaws.com/2006-03-01">
  <LoggingEnabled>
    <TargetBucket>amzn-s3-demo-destination-bucket</TargetBucket>
    <TargetPrefix>logs/</TargetPrefix>
  </LoggingEnabled>
</BucketLoggingStatus>
```

En el siguiente ejemplo se identifica *amzn-s3-demo-destination-bucket* como el bucket de destino, *logs/* como el prefijo y `EventTime` como el formato de clave del objeto de registro.

```
<BucketLoggingStatus xmlns="http://doc.s3.amazonaws.com/2006-03-01">
  <LoggingEnabled>
    <TargetBucket>amzn-s3-demo-destination-bucket</TargetBucket>
    <TargetPrefix>logs/</TargetPrefix>
    <TargetObjectKeyFormat>
      <PartitionedPrefix>
         <PartitionDateSource>EventTime</PartitionDateSource>
      </PartitionedPrefix>
  </TargetObjectKeyFormat>
  </LoggingEnabled>
</BucketLoggingStatus>
```

La cuenta entrega de archivos de registro de S3 escribe y posee los objetos de registro y el propietario del bucket tiene permisos completos sobre los objetos de registro. De forma opcional, puede utilizar concesiones de destino para conceder permisos a otros usuarios para que puedan acceder a los registros. Para obtener más información, consulte PutBucketLogging.

> **Note**
>
> Si el bucket de destino utiliza la configuración de propietario del bucket obligatorio de Object Ownership, no puede usar las concesiones de destino para conceder permisos a otros usuarios. Para conceder permisos a otros, puede utilizar la actualización de la política de bucket en el bucket de destino. Para obtener más información, consulte Permisos para entrega de registros.

Para recuperar la configuración de registro de un bucket, utilice la operación de la API GetBucketLogging.

Para eliminar la configuración de registros, debe enviar la solicitud `PutBucketLogging` con un `BucketLoggingStatus` vacío.

```
<BucketLoggingStatus xmlns="http://doc.s3.amazonaws.com/2006-03-01">
</BucketLoggingStatus>
```

Para habilitar el registro en un bucket, puede utilizar la API de Amazon S3 o las bibliotecas de encapsulamiento de los SDK de AWS.

Uso de los AWS SDK

En los siguientes ejemplos se habilitan los registros en un bucket. Debe crear dos buckets, uno de origen y uno de destino. Los ejemplos actualizan primero la ACL del bucket en el bucket de destino. Conceden al grupo de envío de registros los permisos necesarios para escribir registros en el bucket de destino y, a continuación, habilitan los registros en el bucket de origen.

Estos ejemplos no funcionarán en buckets de destino que utilizan la configuración Aplicada al propietario del bucket obligatorio para la Propiedad de objetos.

Si el bucket de destino utiliza la configuración de propietario del bucket obligatorio de Propiedad de objetos, no se pueden configurar ACL de bucket u objeto. Tampoco puede incluir concesiones de destino en la configuración PutBucketLogging. Debe utilizar una política de bucket para conceder acceso a la entidad principal del servicio de registro (`logging.s3.amazonaws.com`). Para obtener más información, consulte Permisos para entrega de registros.

.NET

AWS SDK for .NET

> **Note**
>
> Hay más información en GitHub. Busque el ejemplo completo y aprenda a configurar y ejecutar en el Repositorio de ejemplos de código de 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();
        }
    }
```

- Para obtener información sobre la API, consulte PutBucketLogging en la Referencia de la API de AWS SDK for .NET.

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());
    }

}
```

Uso de la AWS CLI

Recomendamos que cree un bucket de registro dedicado en cada Región de AWS en la que tenga buckets de S3. A continuación, guarde los registros de acceso de Amazon S3 en ese bucket de S3. Para obtener más información y ejemplos, consulte put-bucket-logging en la Referencia de la AWS CLI.

Si el bucket de destino utiliza la configuración de propietario del bucket obligatorio de Propiedad de objetos, no se pueden configurar ACL de bucket u objeto. Tampoco puede incluir concesiones de destino en la configuración PutBucketLogging. Debe utilizar una política de bucket para conceder acceso a la entidad principal del servicio de registro (`logging.s3.amazonaws.com`). Para obtener más información, consulte Permisos para entrega de registros.

Example — Habilitar registros de acceso con cinco buckets en dos regiones

Para este ejemplo, suponga que tiene los cinco buckets siguientes:

- `1-amzn-s3-demo-bucket1-us-east-1`
- `2-amzn-s3-demo-bucket1-us-east-1`
- `3-amzn-s3-demo-bucket1-us-east-1`
- `1-amzn-s3-demo-bucket1-us-west-2`
- `2-amzn-s3-demo-bucket1-us-west-2`

> **Note**
>
> El último paso del siguiente procedimiento proporciona ejemplos de scripts bash que puede utilizar para crear sus buckets de registro y habilitar el registro de acceso al servidor en estos buckets. Para usar esos scripts, debe crear los archivos `policy.json` y `logging.json`, tal y como se describe en el siguiente procedimiento.

1. Cree dos bucket de destino de registro en las regiones Oeste de EE. UU. (Oregón) y Este de EE. UU. (Norte de Virginia) y asígneles los siguientes nombres:
   - `amzn-s3-demo-bucket1-logs-us-east-1`
   - `amzn-s3-demo-bucket1-logs-us-west-2`
2. Más adelante en estos pasos, habilitará el registro de acceso al servidor de la siguiente manera:

- `1-amzn-s3-demo-bucket1-us-east-1` registra en el bucket de S3 `amzn-s3-demo-bucket1-logs-us-east-1` con el prefijo `1-amzn-s3-demo-bucket1-us-east-1`
- `2-amzn-s3-demo-bucket1-us-east-1` registra en el bucket de S3 `amzn-s3-demo-bucket1-logs-us-east-1` con el prefijo `2-amzn-s3-demo-bucket1-us-east-1`
- `3-amzn-s3-demo-bucket1-us-east-1` registra en el bucket de S3 `amzn-s3-demo-bucket1-logs-us-east-1` con el prefijo `3-amzn-s3-demo-bucket1-us-east-1`
- `1-amzn-s3-demo-bucket1-us-west-2` registra en el bucket de S3 `amzn-s3-demo-bucket1-logs-us-west-2` con el prefijo `1-amzn-s3-demo-bucket1-us-west-2`
- `2-amzn-s3-demo-bucket1-us-west-2` registra en el bucket de S3 `amzn-s3-demo-bucket1-logs-us-west-2` con el prefijo `2-amzn-s3-demo-bucket1-us-west-2`

3. Para cada bucket de registro de destino, conceda permisos para la entrega de registros de acceso al servidor mediante una ACL de bucket o una política de bucket:

   - Actualizar la política de bucket (recomendado): para conceder permisos a la entidad principal del servicio de registro, utilice el siguiente comando `put-bucket-policy`. Reemplace *`amzn-s3-demo-destination-bucket-logs`* con el nombre del bucket de destino.

     ```
     aws s3api put-bucket-policy --bucket amzn-s3-demo-destination-bucket-logs --
     policy file://policy.json
     ```

     `Policy.json` es un documento JSON en la carpeta actual que contiene la siguiente política de bucket. Para utilizar esta política de bucket, reemplace *`user input placeholders`* por su propia información. En la siguiente política, *`amzn-s3-demo-destination-bucket-logs`* es el bucket de destino donde se entregarán los registros de acceso al servidor y *`amzn-s3-demo-source-bucket`* es el bucket de origen. *`SOURCE-ACCOUNT-ID`* es la Cuenta de AWS que posee el bucket de origen.

     ```
     {
         "Version": "2012-10-17",
         "Statement": [
             {
                 "Sid": "S3ServerAccessLogsPolicy",
                 "Effect": "Allow",
                 "Principal": {
                     "Service": "logging.s3.amazonaws.com"
                 },
     ```

```
            "Action": [
                "s3:PutObject"
            ],
            "Resource": "arn:aws:s3:::amzn-s3-demo-destination-bucket-logs/*",
            "Condition": {
                "ArnLike": {
                    "aws:SourceArn": "arn:aws:s3:::amzn-s3-demo-source-bucket"
                },
                "StringEquals": {
                    "aws:SourceAccount": "SOURCE-ACCOUNT-ID"
                }
            }
        }
    ]
}
```

- Actualizar la ACL del bucket: para conceder permisos al grupo de entrega de registros de S3, utilice el siguiente comando `put-bucket-acl`. Reemplace *amzn-s3-demo-destination-bucket-logs* con el nombre del bucket de destino.

```
aws s3api put-bucket-acl --bucket amzn-s3-demo-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. A continuación, cree un archivo `logging.json` que contenga la configuración de registro (según uno de los tres ejemplos siguientes). Tras crear el archivo `logging.json`, puede aplicar la configuración de registro mediante el siguiente comando `put-bucket-logging`. Reemplace *amzn-s3-demo-destination-bucket-logs* con el nombre del bucket de destino.

```
aws s3api put-bucket-logging --bucket amzn-s3-demo-destination-bucket-logs --
bucket-logging-status file://logging.json
```

> **Note**
>
> En lugar de usar este comando `put-bucket-logging` para aplicar la configuración de registro en cada bucket de destino, puede usar uno de los scripts bash que se

proporcionan en el siguiente paso. Para usar esos scripts, debe crear los archivos `policy.json` y `logging.json`, tal y como se describe en este procedimiento.

El archivo `logging.json` es un documento JSON en la carpeta actual que contiene la configuración de su registro. Si un bucket de destino utiliza la configuración de propietario del bucket obligatorio de Object Ownership, la configuración de registro no puede contener concesiones de destino. Para obtener más información, consulte Permisos para entrega de registros.

Example – **`logging.json`** sin concesiones de destino

El siguiente archivo `logging.json` de ejemplo no contiene concesiones de destino. Por lo tanto, puede aplicar esta configuración a un bucket de destino que utilice la configuración Aplicada al propietario del bucket para Propiedad de objetos.

```
{
    "LoggingEnabled": {
        "TargetBucket": "amzn-s3-demo-destination-bucket-logs",
        "TargetPrefix": "amzn-s3-demo-destination-bucket/"
     }
 }
```

Example – **`logging.json`** con subvenciones de destino

El siguiente archivo `logging.json` de ejemplo contiene concesiones de destino.

Si el bucket de destino utiliza la configuración de propietario del bucket obligatorio de propiedad de objetos, no puede incluir las concesiones de destino en la configuración PutBucketLogging. Para obtener más información, consulte Permisos para entrega de registros.

```
{
    "LoggingEnabled": {
        "TargetBucket": "amzn-s3-demo-destination-bucket-logs",
        "TargetPrefix": "amzn-s3-demo-destination-bucket/",
        "TargetGrants": [
             {
```

```
                "Grantee": {
                    "Type": "AmazonCustomerByEmail",
                    "EmailAddress": "user@example.com"
                 },
                "Permission": "FULL_CONTROL"
            }
        ]
    }
}
```

Example – **`logging.json`** con el formato de clave del objeto de registro establecido en la hora del evento de S3

El siguiente archivo `logging.json` cambia el formato de clave del objeto de registro a la hora del evento de S3. Para obtener más información acerca de la configuración del formato de clave de objeto de registro, consulte the section called “¿Cómo habilito la entrega de registros?”.

```
{
  "LoggingEnabled": {
      "TargetBucket": "amzn-s3-demo-destination-bucket-logs",
      "TargetPrefix": "amzn-s3-demo-destination-bucket/",
      "TargetObjectKeyFormat": {
          "PartitionedPrefix": {
              "PartitionDateSource": "EventTime"
          }
       }
    }
}
```

5. Utilice uno de los siguientes scripts bash para agregar el registro de acceso a todos los buckets de la cuenta. Sustituya *`amzn-s3-demo-destination-bucket-logs`* por el nombre de su bucket de destino y sustituya *`us-west-2`* por el nombre de la región en la que se encuentran sus buckets.

> **Note**
>
> Este script solo funciona si todos los buckets están en la misma región. Si tiene buckets en multirregiones, debe ajustar el script.

Example – Conceder acceso con políticas de bucket y agregar registros para los buckets de la cuenta

```
  loggingBucket='amzn-s3-demo-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 – Conceder acceso con ACL de bucket y agregar registros para los buckets de la cuenta

```
  loggingBucket='amzn-s3-demo-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"
```

Verificación de la configuración de los registros de acceso al servidor

Después de habilitar el registro de acceso al servidor, siga los pasos siguientes:

- Acceda al bucket de destino y compruebe que se están entregando los archivos de registro. Una vez configurados los registros de acceso, Amazon S3 comienza inmediatamente a capturar las solicitudes y a registrarlas. Sin embargo, la entrega de los registros al bucket de destino puede tardar unas horas. Para obtener más información, consulte the section called “Los cambios del estado de los registros del bucket surten efecto con el tiempo” y the section called “Envío de archivos de registro de servidor según el mejor esfuerzo”.

  También puede verificar automáticamente la entrega de registros mediante las métricas de solicitud de Amazon S3 y configurar las alarmas de Amazon CloudWatch para estas métricas. Para obtener más información, consulte Monitorización de métricas con Amazon CloudWatch.
- Compruebe que puede abrir y leer el contenido de los archivos de registro.

Para obtener información sobre la solución de problemas del registro de acceso al servidor, consulte Solucionar problemas de registro de acceso al servidor.

Formato de registro de acceso al servidor de Amazon S3

El registro de acceso al servidor proporciona el historial detallado de las solicitudes que se realizan a un bucket de Amazon S3. Puede utilizar los registros de acceso al servidor para los siguientes fines:

- Realizar auditorías de seguridad y acceso
- Obtener información sobre la base de clientes
- Entender la factura de Amazon S3

En esta sección se describe el formato y otros detalles acerca de los archivos de registro de acceso al servidor de Amazon S3.

Los archivos de registro de acceso al servidor consisten en una secuencia de registros delimitados por nueva línea. Cada entrada de registro representa una solicitud y consta de campos delimitados por espacios.

El siguiente es un registro de ejemplo que consta de cinco entradas de registro.

```
79a59df900b949e55d96a1e698fbacedfd6e09d98eacf8f8d5218e7cd47ef2be
 amzn-s3-demo-bucket1 [06/Feb/2019:00:00:38 +0000] 192.0.2.3
 79a59df900b949e55d96a1e698fbacedfd6e09d98eacf8f8d5218e7cd47ef2be 3E57427F3EXAMPLE
 REST.GET.VERSIONING - "GET /amzn-s3-demo-bucket1?versioning HTTP/1.1" 200 - 113 - 7 -
 "-" "S3Console/0.4" - s9lzHYrFp76ZVxRcpX9+5cjAnEH2ROuNkd2BHfIa6UkFVdtjf5mKR3/eTPFvsiP/
XV/VLi31234= SigV4 ECDHE-RSA-AES128-GCM-SHA256 AuthHeader amzn-s3-demo-bucket1.s3.us-
west-1.amazonaws.com TLSV1.2 arn:aws:s3:us-west-1:123456789012:accesspoint/example-AP
 Yes
79a59df900b949e55d96a1e698fbacedfd6e09d98eacf8f8d5218e7cd47ef2be
 amzn-s3-demo-bucket1 [06/Feb/2019:00:00:38 +0000] 192.0.2.3
 79a59df900b949e55d96a1e698fbacedfd6e09d98eacf8f8d5218e7cd47ef2be 891CE47D2EXAMPLE
 REST.GET.LOGGING_STATUS - "GET /amzn-s3-demo-bucket1?logging HTTP/1.1" 200 -
 242 - 11 - "-" "S3Console/0.4" - 9vKBE6vMhrNiWHZmb2L0mXOcqPGzQOI5XLnCtZNPxev+Hf
+7tpT6sxDwDty4LHBUOZJG96N1234= SigV4 ECDHE-RSA-AES128-GCM-SHA256 AuthHeader amzn-s3-
demo-bucket1.s3.us-west-1.amazonaws.com TLSV1.2 - -
79a59df900b949e55d96a1e698fbacedfd6e09d98eacf8f8d5218e7cd47ef2be
 amzn-s3-demo-bucket1 [06/Feb/2019:00:00:38 +0000] 192.0.2.3
 79a59df900b949e55d96a1e698fbacedfd6e09d98eacf8f8d5218e7cd47ef2be A1206F460EXAMPLE
 REST.GET.BUCKETPOLICY - "GET /amzn-s3-demo-bucket1?policy HTTP/1.1" 404
 NoSuchBucketPolicy 297 - 38 - "-" "S3Console/0.4" - BNaBsXZQQDbssi6xMBdBU2sLt
+Yf5kZDmeBUP35sFoKa3sLLeMC78iwEIWxs99CRUrbS4n11234= SigV4 ECDHE-RSA-AES128-GCM-SHA256
 AuthHeader amzn-s3-demo-bucket1.s3.us-west-1.amazonaws.com TLSV1.2 - Yes
79a59df900b949e55d96a1e698fbacedfd6e09d98eacf8f8d5218e7cd47ef2be
 amzn-s3-demo-bucket1 [06/Feb/2019:00:01:00 +0000] 192.0.2.3
 79a59df900b949e55d96a1e698fbacedfd6e09d98eacf8f8d5218e7cd47ef2be 7B4A0FABBEXAMPLE
 REST.GET.VERSIONING - "GET /amzn-s3-demo-bucket1?versioning HTTP/1.1" 200 -
 113 - 33 - "-" "S3Console/0.4" - Ke1bUcazaN1jWuUlPJaxF64cQVpUEhoZKEG/hmy/gijN/
I1DeWqDfFvnpybfEseEME/u7ME1234= SigV4 ECDHE-RSA-AES128-GCM-SHA256 AuthHeader amzn-s3-
demo-bucket1.s3.us-west-1.amazonaws.com TLSV1.2 - -
79a59df900b949e55d96a1e698fbacedfd6e09d98eacf8f8d5218e7cd47ef2be
 amzn-s3-demo-bucket1 [06/Feb/2019:00:01:57 +0000] 192.0.2.3
 79a59df900b949e55d96a1e698fbacedfd6e09d98eacf8f8d5218e7cd47ef2be
 DD6CC733AEXAMPLE REST.PUT.OBJECT s3-dg.pdf "PUT /amzn-s3-demo-bucket1/
s3-dg.pdf HTTP/1.1" 200 - - 4406583 41754 28 "-" "S3Console/0.4" -
 10S62Zv81kBW7BB6SX4XJ48o6kpcl6LPwEoizZQQxJd5qDSCTLX0TgS37kYUBKQW3+bPdrg1234= SigV4
 ECDHE-RSA-AES128-SHA AuthHeader amzn-s3-demo-bucket1.s3.us-west-1.amazonaws.com
 TLSV1.2 - Yes
```

> **Note**
>
> Los campos se pueden establecer en - para indicar que los datos son desconocidos o no están disponibles, o que el campo no se aplica a esta solicitud.

Temas

Registrar campos de registro

En la siguiente lista se describen los campos de entrada de registro.

Propietario del bucket

El ID de usuario canónico del propietario del bucket de origen. El ID de usuario canónico es otra forma del ID de la Cuenta de AWS. Para obtener más información acerca del ID de usuario canónico, consulte Identificadores de la Cuenta de AWS en la Referencia general de AWS. Para obtener información acerca de cómo encontrar el ID de usuario canónico de la cuenta, consulte Búsqueda del ID de usuario canónico para la Cuenta de AWS.

Ejemplo de entrada

```
79a59df900b949e55d96a1e698fbacedfd6e09d98eacf8f8d5218e7cd47ef2be
```

Bucket

El nombre del bucket para el que se procesó la solicitud. Si el sistema recibe un solicitud incorrecta y no puede determinar el bucket, la solicitud no aparecerá en ningún registro de acceso al servidor.

Ejemplo de entrada

```
amzn-s3-demo-bucket1
```

Time

El momento en que se recibió la solicitud; estas fechas y horas están en Hora Universal Coordinada (UTC). El formato, con la terminología `strftime()`, es el siguiente: `[%d/%b/%Y:%H:%M:%S %z]`

Ejemplo de entrada

```
[06/Feb/2019:00:00:38 +0000]
```

IP remota

La dirección IP aparente del solicitante. Los servidores proxy y firewalls intermedios pueden ocultar la dirección IP real de la máquina que realiza la solicitud.

Ejemplo de entrada

```
192.0.2.3
```

Solicitante

El ID de usuario canónico del solicitante o un - para solicitudes no autenticadas. Si el solicitante era un usuario de IAM, este campo devuelve el nombre de usuario de IAM del solicitante junto con la Usuario raíz de la cuenta de AWS a la que pertenece el usuario de IAM. Este identificador es el mismo que se utiliza para el control de acceso.

Ejemplo de entrada

```
79a59df900b949e55d96a1e698fbacedfd6e09d98eacf8f8d5218e7cd47ef2be
```

Si el solicitante utiliza un rol asumido, este campo devuelve el rol de IAM asumido.

Ejemplo de entrada

```
arn:aws:sts::123456789012:assumed-role/roleName/test-role
```

ID de solicitud

Una cadena generada por Amazon S3 para identificar de forma inequívoca cada solicitud.

Ejemplo de entrada

```
3E57427F33A59F07
```

Operation

La operación que se describe aquí se declara como `SOAP.operation`, `REST.HTTP_method.resource_type`, `WEBSITE.HTTP_method.resource_type` o `BATCH.DELETE.OBJECT` o `S3.action.resource_type` para Ciclos de vida y registros.

Ejemplo de entrada

```
REST.PUT.OBJECT
```

Clave

La parte clave (nombre del objeto) de la solicitud.

Ejemplo de entrada

```
/photos/2019/08/puppy.jpg
```

Request-URI

La parte de `Request-URI` del mensaje de solicitud de HTTP.

Ejemplo de entrada

```
"GET /amzn-s3-demo-bucket1/photos/2019/08/puppy.jpg?x-foo=bar HTTP/1.1"
```

Estado HTTP

El código de estado HTTP numérico de la respuesta.

Ejemplo de entrada

```
200
```

Código de error

El Código de error de Amazon S3 o `-` si no se produce ningún error.

Ejemplo de entrada

```
NoSuchBucket
```

Bytes enviados

El número de bytes de respuestas enviados, sin incluir la sobrecarga del protocolo HTTP o `-` en caso de ser cero.

Ejemplo de entrada

```
2662992
```

Tamaño de objeto

El tamaño total del objeto en cuestión.

Ejemplo de entrada

```
3462992
```

Tiempo total

La cantidad de milisegundos que la solicitud estuvo en tránsito desde la perspectiva del servidor. Este valor se mide desde el momento en que se recibe la solicitud hasta el momento en que se envía el último byte de la respuesta. Es posible que las medidas realizadas desde la perspectiva del cliente sean más extensas a causa de la latencia de la red.

Ejemplo de entrada

```
70
```

Tiempo de entrega

La cantidad de milisegundos que tarda Amazon S3 en procesar su solicitud. Este valor se mide desde el momento en que se recibió el último byte de la solicitud hasta el momento en que se envió el primer byte de la respuesta.

Ejemplo de entrada

```
10
```

Referer

El valor del encabezado `Referer` de HTTP, si lo hay. Los agentes de usuario de HTTP (por ejemplo: los navegadores) por lo general configuran este encabezado en la URL de la página enlazada o adjunta cuando realizan una solicitud.

Ejemplo de entrada

```
"http://www.example.com/webservices"
```

User-Agent

El valor del encabezado `User-Agent` de HTTP.

Ejemplo de entrada

```
"curl/7.15.1"
```

ID de versión

El ID de versión en la solicitud o `-` si la operación no toma un parámetro `versionId`.

Ejemplo de entrada

```
3HL4kqtJvjVBH40Nrjfkd
```

ID de host

El `x-amz-id-2` o el ID de la solicitud ampliada de Amazon S3.

Ejemplo de entrada

```
s9lzHYrFp76ZVxRcpX9+5cjAnEH2ROuNkd2BHfIa6UkFVdtjf5mKR3/eTPFvsiP/XV/VLi31234=
```

Versión de firma

La versión de firma, `SigV2` o `SigV4`, que se utilizó para autenticar la solicitud o `-` para las solicitudes no autenticadas.

Ejemplo de entrada

```
SigV2
```

Conjunto de cifrado

Cifrado de Capa de conexión segura (SSL) que se negoció para una solicitud HTTPS o un `-` para HTTP.

Ejemplo de entrada

```
ECDHE-RSA-AES128-GCM-SHA256
```

Tipo de autenticación

Tipo de autenticación de solicitud utilizada: `AuthHeader` para encabezados de autenticación, `QueryString` para cadena de consulta (URL prefirmada) o `-` para solicitudes no autenticadas.

Ejemplo de entrada

```
AuthHeader
```

Encabezado de host

El punto de conexión usado para conectarse a Amazon S3.

Ejemplo de entrada

```
s3.us-west-2.amazonaws.com
```

Algunas regiones anteriores admiten puntos de conexión heredados. Es posible que vea estos puntos de enlace en los registros de acceso al servidor o en los registros de AWS CloudTrail. Para obtener más información, consulte Puntos de conexión heredados. Para obtener una lista completa de las regiones y los puntos de conexión de Amazon S3, consulte Puntos de conexión y cuotas de Amazon S3 en la Referencia general de Amazon Web Services.

Versión de TLS

Versión de Transport Layer Security (TLS) negociada por el cliente. El valor es uno de los siguientes: `TLSv1.1`, `TLSv1.2`, `TLSv1.3` o `-` si no se utilizó TLS.

Ejemplo de entrada

```
TLSv1.2
```

ARN del punto de acceso

El nombre de recurso de Amazon (ARN) del punto de acceso de la solicitud. Si el ARN del punto de acceso está mal formado o no se utiliza, el campo contendrá un `-`. Para obtener más información acerca de los puntos de acceso, consulte Usar puntos de acceso. Para obtener más información acerca de los ARN, consulte Nombre de recurso de Amazon (ARN) en la Guía de referencia de AWS.

Ejemplo de entrada

```
arn:aws:s3:us-east-1:123456789012:accesspoint/example-AP
```

aclRequired

Una cadena que indica si la solicitud requiere una lista de control de acceso (ACL) para la autorización. Si la solicitud requería una ACL para la autorización, la cadena es `Yes`. Si no se requerían ACL, la cadena es `-`. Para obtener más información acerca de las ACL, consulte Información general de las Listas de control de acceso (ACL). Para obtener más información sobre el uso del campo `aclRequired` para desactivar las ACL, consulte Control de la propiedad de los objetos y desactivación de las ACL del bucket.

Ejemplo de entrada

```
Yes
```

Registro adicional para operaciones de copia

Una operación de copia implica un `GET` y un `PUT`. Por esa razón, registramos dos entradas al realizar una operación de copia. En la sección anterior se describen los campos relacionados con la `PUT` parte de la operación. En la siguiente lista se describen los campos del registro relacionados con la parte `GET` de la operación de copia.

Propietario del bucket

El ID de usuario canónico del bucket que almacena el objeto que se copia. El ID de usuario canónico es otra forma del ID de la Cuenta de AWS. Para obtener más información acerca del ID de usuario canónico, consulte Identificadores de la Cuenta de AWS en la Referencia general de AWS. Para obtener información acerca de cómo encontrar el ID de usuario canónico de la cuenta, consulte Búsqueda del ID de usuario canónico para la Cuenta de AWS.

Ejemplo de entrada

```
79a59df900b949e55d96a1e698fbacedfd6e09d98eacf8f8d5218e7cd47ef2be
```

Bucket

El nombre del bucket que almacena el objeto que se copia.

Ejemplo de entrada

```
amzn-s3-demo-bucket1
```

Time

El momento en que se recibió la solicitud; estas fechas y horas están en Hora Universal Coordinada (UTC). El formato, con la terminología `strftime()`, es el siguiente: `[%d/%B/%Y:%H:%M:%S %z]`

Ejemplo de entrada

```
[06/Feb/2019:00:00:38 +0000]
```

IP remota

La dirección IP aparente del solicitante. Los servidores proxy y firewalls intermedios pueden ocultar la dirección IP real de la máquina que realiza la solicitud.

Ejemplo de entrada

```
192.0.2.3
```

Solicitante

El ID de usuario canónico del solicitante o un `-` para solicitudes no autenticadas. Si el solicitante era un usuario de IAM, este campo devolverá el nombre de usuario de IAM del solicitante junto con la Usuario raíz de la cuenta de AWS a la que pertenece el usuario de IAM. Este identificador es el mismo que se utiliza para el control de acceso.

Ejemplo de entrada

```
79a59df900b949e55d96a1e698fbacedfd6e09d98eacf8f8d5218e7cd47ef2be
```

Si el solicitante utiliza un rol asumido, este campo devuelve el rol de IAM asumido.

Ejemplo de entrada

```
arn:aws:sts::123456789012:assumed-role/roleName/test-role
```

ID de solicitud

Una cadena generada por Amazon S3 para identificar de forma inequívoca cada solicitud.

Ejemplo de entrada

```
3E57427F33A59F07
```

Operación

La operación que se indica aquí se declara como `SOAP.operation`, `REST.HTTP_method.resource_type`, `WEBSITE.HTTP_method.resource_type` o `BATCH.DELETE.OBJECT`.

Ejemplo de entrada

```
REST.COPY.OBJECT_GET
```

Clave

La clave (nombre de objeto) del objeto que se copia o "`-`" si la operación no toma un parámetro de clave.

Ejemplo de entrada

```
/photos/2019/08/puppy.jpg
```

Request-URI

La parte de `Request-URI` del mensaje de solicitud de HTTP.

Ejemplo de entrada

```
"GET /amzn-s3-demo-bucket1/photos/2019/08/puppy.jpg?x-foo=bar"
```

Estado HTTP

El código de estado HTTP numérico de la parte `GET` de la operación de copia.

Ejemplo de entrada

```
200
```

Código de error

El Código de error de Amazon S3 de la parte GET de la operación de copia o - si no se produce ningún error.

Ejemplo de entrada

```
NoSuchBucket
```

Bytes enviados

El número de bytes de respuestas enviados, sin incluir la sobrecarga del protocolo HTTP o - en caso de ser cero.

Ejemplo de entrada

```
2662992
```

Tamaño de objeto

El tamaño total del objeto en cuestión.

Ejemplo de entrada

```
3462992
```

Tiempo total

La cantidad de milisegundos que la solicitud estuvo en tránsito desde la perspectiva del servidor. Este valor se mide desde el momento en que se recibe la solicitud hasta el momento en que se envía el último byte de la respuesta. Es posible que las medidas realizadas desde la perspectiva del cliente sean más extensas a causa de la latencia de la red.

Ejemplo de entrada

```
70
```

Tiempo de entrega

La cantidad de milisegundos que tarda Amazon S3 en procesar su solicitud. Este valor se mide desde el momento en que se recibió el último byte de la solicitud hasta el momento en que se envió el primer byte de la respuesta.

Ejemplo de entrada

```
10
```

Referer

El valor del encabezado `Referer` de HTTP, si lo hay. Los agentes de usuario de HTTP (por ejemplo: los navegadores) por lo general configuran este encabezado en la URL de la página enlazada o adjunta cuando realizan una solicitud.

Ejemplo de entrada

```
"http://www.example.com/webservices"
```

User-Agent

El valor del encabezado `User-Agent` de HTTP.

Ejemplo de entrada

```
"curl/7.15.1"
```

ID de versión

El ID de versión del objeto que se copia o `-` si el encabezado `x-amz-copy-source` no especificó un parámetro `versionId` como parte del origen de copia.

Ejemplo de entrada

```
3HL4kqtJvjVBH40Nrjfkd
```

ID de host

El `x-amz-id-2` o el ID de la solicitud ampliada de Amazon S3.

Ejemplo de entrada

```
s9lzHYrFp76ZVxRcpX9+5cjAnEH2ROuNkd2BHfIa6UkFVdtjf5mKR3/eTPFvsiP/XV/VLi31234=
```

Versión de firma

La versión de firma, `SigV2` o `SigV4`, que se utilizó para autenticar la solicitud o `-` para las solicitudes no autenticadas.

Ejemplo de entrada

```
SigV4
```

Conjunto de cifrado

Cifrado de Capa de conexión segura (SSL) que se negoció para una solicitud HTTPS o `-` para HTTP.

Ejemplo de entrada

```
ECDHE-RSA-AES128-GCM-SHA256
```

Tipo de autenticación

Tipo de autenticación de solicitud utilizada: `AuthHeader` para encabezados de autenticación, `QueryString` para cadenas de consulta (URL prefirmadas) o un `-` para solicitudes no autenticadas.

Ejemplo de entrada

```
AuthHeader
```

Encabezado de host

El punto de conexión que se usó para conectarse a Amazon S3.

Ejemplo de entrada

```
s3.us-west-2.amazonaws.com
```

Algunas regiones anteriores admiten puntos de conexión heredados. Es posible que vea estos puntos de enlace en los registros de acceso al servidor o en los registros de AWS CloudTrail. Para obtener más información, consulte Puntos de conexión heredados. Para obtener una lista completa de las regiones y los puntos de conexión de Amazon S3, consulte Puntos de conexión y cuotas de Amazon S3 en la Referencia general de Amazon Web Services.

Versión de TLS

Versión de Transport Layer Security (TLS) negociada por el cliente. El valor es uno de los siguientes: `TLSv1.1`, `TLSv1.2`, `TLSv1.3` o `-` si no se utilizó TLS.

Ejemplo de entrada

```
TLSv1.2
```

ARN del punto de acceso

El nombre de recurso de Amazon (ARN) del punto de acceso de la solicitud. Si el ARN del punto de acceso está mal formado o no se utiliza, el campo contendrá un `-`. Para obtener más información acerca de los puntos de acceso, consulte Usar puntos de acceso. Para obtener más información acerca de los ARN, consulte Nombre de recurso de Amazon (ARN) en la Guía de referencia de AWS.

Ejemplo de entrada

```
arn:aws:s3:us-east-1:123456789012:accesspoint/example-AP
```

aclRequired

Una cadena que indica si la solicitud requiere una lista de control de acceso (ACL) para la autorización. Si la solicitud requería una ACL para la autorización, la cadena es `Yes`. Si no se requerían ACL, la cadena es `-`. Para obtener más información acerca de las ACL, consulte Información general de las Listas de control de acceso (ACL). Para obtener más información sobre el uso del campo `aclRequired` para desactivar las ACL, consulte Control de la propiedad de los objetos y desactivación de las ACL del bucket.

Ejemplo de entrada

```
Yes
```

Información de registro de acceso personalizada

Puede incluir información personalizada que se almacenará en el registro de registro de acceso para una solicitud. Para ello, agregue un parámetro de cadena de consulta personalizado a la URL de la solicitud. Amazon S3 pasa por alto los parámetros de cadena de consulta que empiezan con

`x-`, pero los incluye en la entrada de registro de acceso para la solicitud, como parte del campo `Request-URI` de la entrada de registro.

Por ejemplo, una `GET` solicitud de `"s3.amazonaws.com/amzn-s3-demo-bucket1/photos/2019/08/puppy.jpg?x-user=johndoe"` funciona igual que la solicitud de `"s3.amazonaws.com/amzn-s3-demo-bucket1/photos/2019/08/puppy.jpg"`, excepto que la `"x-user=johndoe"` cadena se incluye en el `Request-URI` campo para el historial de registro asociado. Esta funcionalidad está disponible en la interfaz de REST únicamente.

Consideraciones de programación para el formato de registro de acceso al servidor extensible

Ocasionalmente podríamos ampliar el formato de registro de acceso al agregar nuevos campos al final de cada línea. Por lo tanto, asegúrese de que cualquier código que analiza los registros de acceso al servidor pueda manejar los campos finales que podría no entender.

Eliminación de archivos de registro de Amazon S3

Un bucket de Amazon S3 con registro de acceso al servidor habilitado puede acumular muchos objetos de registro de servidor a lo largo del tiempo. Es posible que la aplicación necesite estos registros de acceso durante un periodo específico después de crearlos y, después de eso, es posible que desee eliminarlos. Puede utilizar la configuración de ciclo de vida de Amazon S3 para establecer reglas para que Amazon S3 ponga automáticamente en cola estos objetos y los elimine al final de su ciclo de vida.

Puede definir una configuración de ciclo de vida para un subconjunto de objetos del bucket de S3 mediante un prefijo compartido. Si especificó un prefijo en su configuración de registro de acceso al servidor, puede establecer una regla de configuración de ciclo de vida para eliminar los objetos de registro que tienen ese prefijo.

Por ejemplo, supongamos que sus objetos de registro tienen el prefijo `logs/`. Puede establecer una regla de configuración de ciclo de vida para eliminar todos los objetos del bucket que tengan el prefijo `logs/` después de un periodo de tiempo especificado.

Para obtener más información acerca de la configuración del ciclo de vida, consulte Administración del ciclo de vida del almacenamiento.

Para obtener información general sobre el registro de acceso al servidor, consulte Registro de solicitudes con registro de acceso al servidor.

Uso de los registros de acceso al servidor de Amazon S3 para identificar solicitudes

Puede identificar las solicitudes de Amazon S3 mediante los registros de acceso al servidor de Amazon S3.

> **Note**
>
> - Para identificar solicitudes de Amazon S3, le recomendamos que utilice eventos de datos de AWS CloudTrail en lugar de los registros de acceso al servidor de Amazon S3. Los eventos de datos de CloudTrail son más fáciles de configurar y contienen más información. Para obtener más información, consulte Identificación de solicitudes de Amazon S3 mediante CloudTrail.
> - Según el número de solicitudes de acceso que obtenga, analizar sus registros puede requerir más recursos o tiempo que usar eventos de datos de CloudTrail.

Temas

Consultar los registros de acceso para solicitudes mediante Amazon Athena

Puede identificar las solicitudes de Amazon S3 con los registros de acceso de Amazon S3 mediante Amazon Athena.

Amazon S3 almacena los registros de acceso del servidor como objetos en un bucket de S3. Suele ser más fácil utilizar una herramienta que pueda analizar los registros en Amazon S3. Athena admite el análisis de objetos de S3 y se puede utilizar para consultar los registros de acceso de Amazon S3.

Example

El siguiente ejemplo muestra cómo puede consultar los registros de acceso al servidor de Amazon S3 en Amazon Athena. Reemplace los *`user input placeholders`* utilizados en los siguientes ejemplos con su propia información.

> **Note**
>
> Para especificar una ubicación de Amazon S3 en una consulta de Athena, debe proporcionar un URI de S3 para el bucket donde se van a entregar sus registros. Este URI debe incluir el nombre y el prefijo del bucket en el siguiente formato: `s3://amzn-s3-demo-bucket1-logs/prefix/`

1. Abra la consola de Athena en https://console.aws.amazon.com/athena/.
2. En el Editor de consultas, ejecute un comando similar al siguiente. Sustituya *s3_access_logs_db* por el nombre que desea asignar a la base de datos.

```
CREATE DATABASE s3_access_logs_db
```

> **Note**
>
> Una práctica recomendada es la creación de la base de datos en la misma Región de AWS que el bucket de S3.

3. En el editor de consultas, ejecute un comando similar al siguiente para crear un esquema de tabla en la base de datos que creó en el paso 2. Sustituya *s3_access_logs_db.mybucket_logs* por el nombre que desea asignar a la tabla. Los valores con los tipos de datos `STRING` y `BIGINT` son las propiedades del registro de acceso. Puede consultar estas propiedades en Athena. Para `LOCATION`, introduzca el bucket de S3 y la ruta del prefijo como se indicó anteriormente.

```
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://amzn-s3-demo-bucket1-logs/prefix/'
```

4. En el panel de navegación, en Database (Base de datos), elija la base de datos.
5. En Tables (Tablas), elija Preview table (Vista previa de tabla) junto al nombre de la tabla.

   En el panel Results (Resultados), debería ver los datos de los registros de acceso del servidor, como `bucketowner`, `bucket`, `requestdatetime`, etc. Esto significa que ha creado correctamente la tabla de Athena. Ahora puede consultar los registros de acceso al servidor de Amazon S3.

Example — Mostrar quién eliminó un objeto y cuándo (marca temporal, dirección IP y usuario de IAM)

```
SELECT requestdatetime, remoteip, requester, key
FROM s3_access_logs_db.mybucket_logs
```

```
WHERE key = 'images/picture.jpg' AND operation like '%DELETE%';
```

Example — Mostrar todas las operaciones realizadas por un usuario de IAM

```
SELECT *
FROM s3_access_logs_db.mybucket_logs
WHERE requester='arn:aws:iam::123456789123:user/user_name';
```

Example — Mostrar todas las operaciones realizadas en un objeto en un periodo de tiempo específico

```
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 — Mostrar la cantidad de datos transferidos a una dirección IP específica en un periodo de tiempo específico

```
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**
>
> A fin de reducir el tiempo que se retienen sus registros, puede crear una configuración del ciclo de vida de S3 para el bucket de registros de acceso al servidor. Cree reglas de configuración del ciclo de vida para eliminar los archivos de registro periódicamente. Esto

reduce la cantidad de datos que Athena analiza para cada consulta. Para obtener más información, consulte Configuración de un ciclo de vida en un bucket.

Identificación de solicitudes de la versión 2 de firma mediante registros de acceso de Amazon S3

La compatibilidad de Amazon S3 con Signature Version 2 va a finalizar (esta característica quedará obsoleta). Cuando esto suceda, Amazon S3 dejará de aceptar solicitudes que utilicen Signature Version 2, y todas las solicitudes deberán firmarse con Signature Version 4. Puede identificar las solicitudes de Signature Version 2 utilizando los registros de acceso de Amazon S3.

Note

Para identificar solicitudes de Signature Version 2, le recomendamos que utilice eventos de datos de AWS CloudTrail en lugar de los registros de acceso al servidor de Amazon S3. Los eventos de datos de CloudTrail son más fáciles de configurar y contienen más información que los registros de acceso del servidor. Para obtener más información, consulte Identificación de solicitudes de firma de Amazon S3 versión 2 mediante CloudTrail.

Example — Mostrar todos los solicitantes que envían tráfico de la versión 2 de la firma

```
SELECT requester, sigv, Count(sigv) as sigcount
FROM s3_access_logs_db.mybucket_logs
GROUP BY requester, sigv;
```

Identificación de solicitudes de acceso a objetos mediante los registros de acceso de Amazon S3

Puede usar consultas en registros de acceso al servidor de Amazon S3 para identificar las solicitudes de acceso a objetos de Amazon S3 para operaciones como `GET`, `PUT` y `DELETE` y obtener información sobre esas solicitudes.

El siguiente ejemplo de consulta de Amazon Athena muestra cómo obtener todas las solicitudes de objetos `PUT` para Amazon S3 desde un registro de acceso al servidor.

Example — Mostrar todos los solicitantes que envían solicitudes de objetos **PUT** en un periodo determinado

```
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')
```

El siguiente ejemplo de consulta de Amazon Athena muestra cómo obtener todas las solicitudes de objetos GET para Amazon S3 desde el registro de acceso al servidor.

Example — Mostrar todos los solicitantes que envían solicitudes de objetos **GET** en un periodo determinado

```
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')
```

La siguiente consulta de ejemplo de Amazon Athena muestra cómo obtener todas las solicitudes anónimas realizadas a los buckets de S3 desde el registro de acceso al servidor.

Example — Mostrar todos los solicitantes anónimos que hacen solicitudes a un bucket durante un periodo determinado

```
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 siguiente consulta de Amazon Athena muestra cómo identificar todas las solicitudes realizadas a los buckets de S3 que requerían una lista de control de acceso (ACL) para la autorización. Puede usar esta información para migrar esos permisos de ACL a las políticas de bucket adecuadas y desactivar las ACL. Una vez que haya creado estas políticas de bucket, puede desactivar las ACL de estos buckets. Para obtener más información acerca de la desactivación de las ACL, consulte Requisitos previos para desactivar las ACL.

Example — Identificar todas las solicitudes que requerían una ACL para la autorización

```
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

- Puede modificar el intervalo de fechas según sea necesario.
- Estos ejemplos de consulta también pueden ser útiles para la monitorización de la seguridad. Puede revisar los resultados de las llamadas a las operaciones `PutObject` o `GetObject` desde solicitantes o direcciones IP inesperados o no autorizados con el fin de identificar cualquier solicitud anónima que se realice a los buckets.
- Esta consulta solo recupera información de la hora a la que se habilitó el registro.
- Si utiliza registros de AWS CloudTrail, consulte Identificación del acceso a objetos S3 mediante CloudTrail.

Monitorización de métricas con Amazon CloudWatch

Las métricas de Amazon CloudWatch para Amazon S3 pueden ayudarle a comprender y mejorar el rendimiento de las aplicaciones que utilizan Amazon S3. Existen varias formas de utilizar CloudWatch con Amazon S3.

Métricas de almacenamiento diario para buckets

Puede monitorear el almacenamiento de buckets mediante CloudWatch, que recopila y procesa datos de almacenamiento de Amazon S3 en métricas diarias legibles. El informe de estas métricas de almacenamiento para Amazon S3 se realiza una vez al día, y se facilita a todos los clientes sin coste adicional.

Métricas de solicitudes

Permiten monitorear las solicitudes de Amazon S3 para identificar rápidamente los problemas operativos y actuar en consecuencia. Las métricas están disponibles en intervalos de 1 minuto después de un breve periodo de latencia para procesarlas. Estas métricas de CloudWatch se facturan al mismo precio que las métricas personalizadas de Amazon CloudWatch. Para obtener más información sobre los precios de CloudWatch, consulte Precios de Amazon CloudWatch. Para saber cómo activar la obtención de estas métricas, consulte Configuraciones de métricas de CloudWatch.

Cuando están habilitadas, se informa de las métricas de solicitudes para todas las operaciones con objetos. De forma predeterminada, estas métricas de 1 minuto están disponibles en el nivel del bucket de Amazon S3. También puede definir un filtro para las métricas utilizando un prefijo compartido, etiqueta de objeto o punto de acceso:

- Punto de acceso: los puntos de acceso se denominan puntos de enlace de red que se adjunta a los buckets y simplifican la administración del acceso a los datos a escala para los conjuntos de datos compartidos en S3. Con el filtro de puntos de acceso, puede obtener información sobre el uso del punto de acceso. Para obtener más información acerca de los puntos de acceso, consulte Monitorización y registro de puntos de acceso.
- Prefijo: aunque el modelo de datos de Amazon S3 es una estructura plana, puede inferir la jerarquía con un prefijo. Un prefijo es similar al nombre de un directorio que permite agrupar objetos similares en un bucket. La consola de S3 admite estos prefijos con el concepto de carpetas. Si filtra por prefijo, los objetos que tengan el mismo prefijo se incluyen en la configuración de métricas. Para obtener más información acerca de los prefijos, consulte Organizar objetos con prefijos.
- Etiquetas: puede agregar etiquetas, que son pares de nombre de clave-valor, a los objetos. Las etiquetas le permiten encontrar y organizar los objetos fácilmente. También puede utilizar etiquetas como filtro para las configuraciones de métricas de modo que solo se incluyan en la configuración de métricas. Para obtener más información acerca de las etiquetas de objeto, consulte Categorización del almacenamiento mediante etiquetas.

Para alinear estas métricas a aplicaciones de negocios, flujos de trabajo u organizaciones internas específicos, puede filtrar un prefijo compartido, etiqueta de objeto o punto de acceso.

Métricas de replicación

Métricas de replicación: monitorice el número total de operaciones de la API de S3 que están pendientes de replicación, el tamaño total de los objetos pendientes de replicación, el tiempo máximo de replicación en la Región de AWSde destino y el número total de operaciones que no se ha podido replicar. Las reglas de replicación que tengan activado el control de tiempo de replicación de S3 (S3 RTC) o las métricas de replicación de S3 habilitadas publicarán métricas de replicación.

Para obtener más información, consulte Monitoreo del progreso con métricas de replicación y notificaciones de eventos de S3 o Cumplimiento de los requisitos de conformidad mediante el control de tiempo de replicación de S3 (S3 RTC).

Métricas de Amazon S3 Storage Lens

Puede publicar métricas de actividad y uso de S3 Storage Lens en Amazon CloudWatch para crear una vista unificada del estado operativo en los paneles de CloudWatch. Las métricas de S3 Storage Lens están disponibles en el espacio de nombres de `AWS/S3/Storage-Lens`. La opción de publicación de CloudWatch está disponible para los paneles de S3 Storage Lens actualizados a métricas y recomendaciones avanzadas. Puede habilitar la opción de publicación de CloudWatch para una configuración de panel nueva o existente en S3 Storage Lens.

Para obtener más información, consulte Monitoreo de métricas de S3 Storage Lens en CloudWatch.

Todas las estadísticas de CloudWatch se retienen durante un periodo de 15 meses, lo que le permite tener acceso a información histórica y obtener una mejor perspectiva sobre el rendimiento de su aplicación o servicio web. Para obtener más información acerca de CloudWatch, consulte ¿Qué es Amazon CloudWatch? en la Guía del usuario de Amazon CloudWatch. Es posible que necesite algunas configuraciones adicionales para las alarmas de CloudWatch, en función de sus casos de uso. Por ejemplo, puede utilizar una expresión matemática de métrica para crear una alarma. Para obtener más información, consulte Use CloudWatch metrics (Uso de las métricas de CloudWatch), Use metric math (Uso de expresiones matemáticas de métricas), Using Amazon CloudWatch alarms (Uso de alarmas de Amazon CloudWatch) y Create a CloudWatch alarm based on a metric math expression (Creación de una alarma de CloudWatch basada en una expresión matemática de métrica) en la Guía del usuario de Amazon CloudWatch.

Entrega de métricas de CloudWatch en la medida de lo posible

Las métricas de CloudWatch se entregan en la medida que es posible. La mayoría de solicitudes de un objeto de Amazon S3 que tienen métricas de solicitudes se derivan en el envío de un punto de datos a CloudWatch.

La integridad y la puntualidad de las métricas no están garantizadas. Es posible que el punto de datos de una solicitud determinada se envíe con una marca temporal posterior al momento en el que la solicitud se ha procesado realmente. Es posible que el punto de datos durante un minuto se retrase antes de estar disponible a través de CloudWatch, o puede que no se entregue en absoluto. Las métricas de solicitudes de CloudWatch le dan una idea de la naturaleza del tráfico al que se enfrenta su bucket en tiempo casi real. No pretende ser un recuento completo de todas las solicitudes.

Dado que esta característica funciona en la medida de lo posible, los informes disponibles en el panel de Administración de facturación y costos podrían incluir una o varias solicitudes de acceso que no aparecen en las métricas del bucket.

Para obtener más información, consulte los siguientes temas.

Temas

- Métricas y dimensiones
- Acceso a métricas de CloudWatch
- Configuraciones de métricas de CloudWatch

Métricas y dimensiones

En las siguientes tablas se detallan las métricas y las dimensiones que Amazon S3 envía a Amazon CloudWatch.

Entrega de métricas de CloudWatch en la medida de lo posible

Las métricas de CloudWatch se entregan en la medida que es posible. La mayoría de solicitudes de un objeto de Amazon S3 que tienen métricas de solicitudes se derivan en el envío de un punto de datos a CloudWatch.

La integridad y la puntualidad de las métricas no están garantizadas. Es posible que el punto de datos de una solicitud determinada se envíe con una marca temporal posterior al momento en el que la solicitud se ha procesado realmente. Es posible que el punto de datos durante un minuto

se retrase antes de estar disponible a través de CloudWatch, o puede que no se entregue en absoluto. Las métricas de solicitudes de CloudWatch le dan una idea de la naturaleza del tráfico al que se enfrenta su bucket en tiempo casi real. No pretende ser un recuento completo de todas las solicitudes.

Dado que esta característica funciona en la medida de lo posible, los informes disponibles en el panel de Administración de facturación y costos podrían incluir una o varias solicitudes de acceso que no aparecen en las métricas del bucket.

Temas

Métricas de almacenamiento diario de Amazon S3 para buckets en CloudWatch

El espacio de nombres `AWS/S3` incluye las siguientes métricas diarias de almacenamiento para los buckets.

Métrica	Descripción
`BucketSizeBytes`	La cantidad de datos en bytes que se almacena en un bucket en las siguientes clases de almacenamiento: • S3 Standard (`STANDARD`) • S3 Intelligent-Tiering (`INTELLIGENT_TIERING`) • S3 Standard-Infrequent Access (`STANDARD_IA`) • S3 One Zone-Infrequent Access (`ONEZONE_IA`)

Métrica	Descripción
	• Almacenamiento de redundancia reducida (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`) Este valor se calcula sumando el tamaño de todos los objetos y metadatos (como los nombres de bucket) en el bucket (tanto los objetos actuales como los no actuales), incluido el tamaño de todas las partes correspondientes a todas las cargas multiparte incompletas en el bucket. **Note** La clase de almacenamiento S3 Express One Zone solo está disponible para buckets de directorio. Filtros de tipos de almacenamiento válidos (consulte la dimensión `StorageType`): • S3 Standard: `StandardStorage` • S3 Intelligent-Tiering: `IntelligentTieringFAStorage`, `IntelligentTieringIAStorage`, `IntelligentTieringAAStorage`, `IntelligentTieringAIAStorage`, `IntelligentTieringDAAStorage` • S3 Standard-Infrequent Access: `StandardIAStorage`, `StandardIASizeOverhead`, `StandardIAObjectOverhead` • S3 One Zone-Infrequent Access: `OneZoneIAStorage`, `OneZoneIASizeOverhead` • Almacenamiento de redundancia reducida (RRS): `ReducedRedundancyStorage`

Métrica	Descripción
	• S3 Glacier Instant Retrieval: `GlacierInstantRetrievalSizeOverhead` , `GlacierInstantRetrievalStorage` • S3 Glacier Flexible Retrieval: `GlacierStorage` , `GlacierStagingStorage` , `GlacierObjectOverhead` , `GlacierS3ObjectOverhead` • S3 Glacier Deep Archive: `DeepArchiveStorage` , `DeepArchiveObjectOverhead` , `DeepArchiveS3ObjectOverhead` , `DeepArchiveStagingStorage` • S3 Express One Zone: `ExpressOneZoneStorage` Unidades: bytes Estadísticas válidas: Average Para obtener más información acerca de las dimensiones `StorageType` , consulte the section called “Dimensiones de Amazon S3 en CloudWatch”.
`NumberOfObjects`	Número total de objetos almacenados en un bucket de uso general para todas las clases de almacenamiento. Este valor se calcula contando todos los objetos en el bucket, que incluye objetos actuales y no actuales, marcadores de eliminación y el número total de partes correspondientes a todas las cargas de multiparte incompletas en el bucket. Para los bucket de directorio con objetos en la clase de almacenamiento S3 Express One Zone, este valor se calcula contando todos los objetos del bucket, pero no incluye las cargas múltiples incompletas en el bucket. Filtros de tipos de almacenamiento válidos: `AllStorageTypes` (consulte la dimensión `StorageType`) Unidades: recuento Estadísticas válidas: Average.

Métricas de solicitud de Amazon S3 en CloudWatch

El espacio de nombres `AWS/S3` incluye las siguientes métricas de solicitudes. Estas métricas incluyen las solicitudes no facturables (en el caso de las solicitudes `GET` de `CopyObject` y Replicación).

> **Note**
>
> Las métricas de solicitud de Amazon S3 en CloudWatch no son compatibles con los buckets de directorio.

Métrica	Descripción
`AllRequests`	Número total de solicitudes HTTP realizadas en un bucket de Amazon S3, independientemente del tipo. Si usa una configuración de métricas con un filtro, esta métrica devuelve únicamente las solicitudes HTTP que cumplen los requisitos del filtro. Unidades: recuento Estadísticas válidas: Sum
`GetRequests`	Número de solicitudes HTTP `GET` realizadas para los objetos de un bucket de Amazon S3. No incluye las operaciones de lista. Esta métrica se incrementa para el origen de cada solicitud `CopyObject` . Unidades: recuento Estadísticas válidas: Sum **Note** Las solicitudes relacionadas con listas paginadas, como ListMultipartUploads, ListParts y ListObjectVersions, entre otras, no se incluyen en esta métrica.

Métrica	Descripción
`PutRequests`	Número de solicitudes HTTP `PUT` realizadas para los objetos de un bucket de Amazon S3. Esta métrica se incrementa para el destino de cada solicitud `CopyObject` . Unidades: recuento Estadísticas válidas: Sum
`DeleteRequests`	Número de solicitudes HTTP `DELETE` realizadas para los objetos de un bucket de Amazon S3. Esta métrica también incluye las solicitudes DeleteObjects. Esta métrica indica el número de solicitudes realizadas, no el número de objetos eliminados. Unidades: recuento Estadísticas válidas: Sum
`HeadRequests`	Número de solicitudes HTTP `HEAD` realizadas a un bucket de Amazon S3. Unidades: recuento Estadísticas válidas: Sum
`PostRequests`	Número de solicitudes HTTP `POST` realizadas a un bucket de Amazon S3. Unidades: recuento Estadísticas válidas: Sum **Note** Las solicitudes DeleteObjects y SelectObjectContent no se incluyen en esta métrica.

Métrica	Descripción
`SelectRequests`	Número de solicitudes SelectObjectContent de Amazon S3 realizadas para los objetos de un bucket de Amazon S3. Unidades: recuento Estadísticas válidas: Sum
`SelectBytesScanned`	Es el número de bytes de datos analizados con solicitudes SelectObjectContent de Amazon S3 en un bucket de Amazon S3. Unidades: bytes Estadísticas válidas: Average (bytes por solicitud), Sum (bytes por periodo), Sample Count, Min, Max (igual que p100) y cualquier percentil entre p 0,0 y p 99,9.
`SelectBytesReturned`	Es el número de bytes de datos devueltos con solicitudes SelectObjectContent de Amazon S3 en un bucket de Amazon S3. Unidades: bytes Estadísticas válidas: Average (bytes por solicitud), Sum (bytes por periodo), Sample Count, Min, Max (igual que p100) y cualquier percentil entre p 0,0 y p 99,9.
`ListRequests`	Número de solicitudes HTTP que muestran el contenido de un bucket. Unidades: recuento Estadísticas válidas: Sum
`BytesDownloaded`	Número de bytes descargados para las solicitudes realizadas a un bucket de Amazon S3 en las que la respuesta contiene un cuerpo. Unidades: bytes Estadísticas válidas: Average (bytes por solicitud), Sum (bytes por periodo), Sample Count, Min, Max (igual que p100) y cualquier percentil entre p 0,0 y p 99,9.

Métrica	Descripción
`BytesUploaded`	Número de bytes cargados para las solicitudes realizadas a un bucket de Amazon S3 en las que la solicitud incluye un cuerpo. Unidades: bytes Estadísticas válidas: Average (bytes por solicitud), Sum (bytes por periodo), Sample Count, Min, Max (igual que p100) y cualquier percentil entre p 0,0 y p 99,9.
`4xxErrors`	Es el número de solicitudes con el código de estado de error del cliente HTTP 4xx realizadas a un bucket de Amazon S3 con un valor de 0 o 1. La estadística Average muestra el porcentaje de error y la estadística muestra las veces que se ha producido el error durante cada periodo. Unidades: recuento Estadísticas válidas: Average (informes por solicitud), Sum (informes por periodo), Min, Max, Sample Count.
`5xxErrors`	Es el número de solicitudes con el código de estado de error del servidor HTTP 5xx realizadas en un bucket de Amazon S3 con un valor de 0 o 1. La estadística Average muestra el porcentaje de error y la estadística muestra las veces que se ha producido el error durante cada periodo. Unidades: recuento Estadísticas válidas: Average (informes por solicitud), Sum (informes por periodo), Min, Max, Sample Count.
`FirstByte Latency`	Tiempo por solicitud desde que un bucket de Amazon S3 recibe la solicitud completa hasta que empieza a devolverse una respuesta. Unidades: milisegundos Estadísticas válidas: Average, Sum, Min, Max (igual que p100), Sample Count y cualquier percentil entre p 0,0 y p100.

Métrica	Descripción
`TotalRequestLatency`	Tiempo por solicitud transcurrido desde que se recibe el primer byte hasta que se envía el último byte a un bucket de Amazon S3. Esta métrica incluye el tiempo que se tarda en recibir el cuerpo de la solicitud y en enviar el cuerpo de la respuesta, que no se incluye en `FirstByteLatency` . Unidades: milisegundos Estadísticas válidas: Average, Sum, Min, Max (igual que p100), Sample Count y cualquier percentil entre p 0,0 y p100.

Métricas de replicación de S3 en CloudWatch

Puede monitorear el progreso de la replicación con métricas de replicación de S3 mediante el seguimiento de bytes y operaciones pendientes y latencia de replicación. Para obtener más información, consulte Monitoreo del progreso con métricas de replicación.

> **Note**
>
> Puede habilitar alarmas para sus métricas de replicación en Amazon CloudWatch. Al configurar alarmas para las métricas de replicación, establezca el campo Missing data treatment (Tratamiento si faltan datos) en Treat missing data as ignore (maintain the alarm state) (Omitir los datos que faltan [mantener el estado de alarma]).

Métrica	Descripción
`ReplicationLatency`	Número máximo de segundos de retraso de la Región de AWS de destino de la replicación respecto a la Región de AWS de origen para una regla de replicación determinada. Unidades: segundos Estadísticas válidas: Max

Métrica	Descripción
`BytesPendingReplication`	Número total de bytes de objetos pendientes de replicación para una regla de replicación determinada. Unidades: bytes Estadísticas válidas: Max
`OperationsPendingReplication`	Número de operaciones pendientes de replicación para una regla de replicación determinada. Unidades: recuento Estadísticas válidas: Max
`OperationsFailedReplication`	Número de operaciones cuya replicación ha fallado para una regla de replicación determinada. Unidades: recuento Estadísticas válidas: suma (número total de operaciones fallidas), promedio (porcentaje de errores), recuento de muestras (número total de operaciones de replicación)

Métricas de S3 Storage Lens en CloudWatch

Puede publicar métricas de actividad y uso de S3 Storage Lens en Amazon CloudWatch para crear una vista unificada del estado operativo en los paneles de CloudWatch. Las métricas de S3 Storage Lens se publican en el espacio de nombres `AWS/S3/Storage-Lens` en CloudWatch. La opción de publicación de CloudWatch está disponible para los paneles de S3 Storage Lens actualizados a métricas y recomendaciones avanzadas.

Para obtener una lista de las métricas de S3 Storage Lens publicadas en CloudWatch, consulte Glosario de métricas de Amazon S3 Storage Lens. Para obtener una lista completa de las dimensiones, consulte Dimensiones.

Métricas de solicitud de S3 Object Lambda en CloudWatch

S3 Object Lambda incluye las siguientes métricas de solicitudes.

Métrica	Descripción
`AllRequests`	Número total de solicitudes HTTP realizadas a un bucket de Amazon S3 utilizando un punto de acceso de Object Lambda. Unidades: recuento Estadísticas válidas: Sum
`GetRequests`	Número de solicitudes HTTP `GET` realizadas para los objetos utilizando un punto de acceso de Object Lambda. Esta métrica no incluye las operaciones de lista. Unidades: recuento Estadísticas válidas: Sum
`BytesUploaded`	Número de bytes cargados en un bucket de Amazon S3 utilizando un punto de acceso de Object Lambda, donde la solicitud incluye un cuerpo. Unidades: bytes Estadísticas válidas: Average (bytes por solicitud), Sum (bytes por periodo), Sample Count, Min, Max (igual que p100) y cualquier percentil entre p 0,0 y p 99,9.
`PostRequests`	Número de solicitudes HTTP `POST` realizadas a un bucket de Amazon S3 utilizando un punto de acceso de Object Lambda. Unidades: recuento Estadísticas válidas: Sum
`PutRequests`	Número de solicitudes HTTP `PUT` realizadas a objetos en un bucket de Amazon S3 utilizando un punto de acceso de Object Lambda. Unidades: recuento Estadísticas válidas: Sum

Métrica	Descripción
`DeleteRequests`	Número de solicitudes HTTP `DELETE` realizadas a objetos en un bucket de Amazon S3 utilizando un punto de acceso de Object Lambda. Esta métrica también incluye las solicitudes DeleteObjects. Esta métrica indica el número de solicitudes realizadas, no el número de objetos eliminados. Unidades: recuento Estadísticas válidas: Sum
`BytesDownloaded`	Número de bytes descargados para las solicitudes realizadas a un bucket de Amazon S3 utilizando un punto de acceso de Object Lambda, donde la respuesta contiene un cuerpo. Unidades: bytes Estadísticas válidas: Average (bytes por solicitud), Sum (bytes por periodo), Sample Count, Min, Max (igual que p100) y cualquier percentil entre p 0,0 y p 99,9.
`FirstByte Latency`	Tiempo por solicitud desde que se recibe la solicitud completa en un bucket de Amazon S3 a través de un punto de acceso de Object Lambda hasta que se empieza a devolver la respuesta. Esta métrica depende del tiempo de ejecución de la función AWS Lambda para transformar el objeto antes de que la función devuelva los bytes al punto de acceso de Object Lambda. Unidades: milisegundos Estadísticas válidas: Average, Sum, Min, Max (igual que p100), Sample Count y cualquier percentil entre p 0,0 y p100.

Métrica	Descripción
`TotalRequestLatency`	Tiempo por solicitud transcurrido desde que se recibió el primer byte hasta que se envió el último byte a un punto de acceso de Object Lambda. Esta métrica incluye el tiempo que se tarda en recibir el cuerpo de la solicitud y en enviar el cuerpo de la respuesta, que no se incluye en `FirstByteLatency` . Unidades: milisegundos Estadísticas válidas: Average, Sum, Min, Max (igual que p100), Sample Count y cualquier percentil entre p 0,0 y p100.
`HeadRequests`	Número de solicitudes HTTP `HEAD` realizadas a un bucket de Amazon S3 utilizando un punto de acceso de Object Lambda. Unidades: recuento Estadísticas válidas: Sum
`ListRequests`	Número de solicitudes de HTTP `GET` que muestran el contenido de un bucket de Amazon S3. Esta métrica incluye las operaciones `ListObjects` y `ListObjectsV2` . Unidades: recuento Estadísticas válidas: Sum
`4xxErrors`	El número de solicitudes con el código de estado de error del cliente HTTP 4xx realizadas a un bucket de Amazon S3 mediante un punto de acceso de Object Lambda con un valor de 0 o 1. La estadística Average muestra el porcentaje de error y la estadística muestra las veces que se ha producido el error durante cada periodo. Unidades: recuento Estadísticas válidas: Average (informes por solicitud), Sum (informes por periodo), Min, Max, Sample Count.

Métrica	Descripción
`5xxErrors`	Es el número de solicitudes con el código de estado de error del servidor HTTP 5xx realizadas a un bucket de Amazon S3 utilizando un punto de acceso de Object Lambda con un valor de 0 o 1. La estadística Average muestra el porcentaje de error y la estadística muestra las veces que se ha producido el error durante cada periodo. Unidades: recuento Estadísticas válidas: Average (informes por solicitud), Sum (informes por periodo), Min, Max, Sample Count.
`ProxiedRequests`	Número de solicitudes HTTP a un punto de acceso de Object Lambda que devuelven la respuesta estándar de la API de Amazon S3. (Estas solicitudes no tienen configurada una función de Lambda). Unidades: recuento Estadísticas válidas: Sum
`InvokedLambda`	El número de solicitudes HTTP a un objeto S3 en el que se invocó una función de Lambda. Unidades: recuento Estadísticas válidas: Sum
`LambdaResponseRequests`	Número de solicitudes `WriteGetObjectResponse` realizadas por la función de Lambda. Esta métrica se aplica únicamente a solicitudes `GetObject` .
`LambdaResponse4xx`	Es el número de errores del cliente HTTP 4xx que se produce al llamar a `WriteGetObjectResponse` desde una función de Lambda. Esta métrica proporciona la misma información que `4xxErrors` , pero solo para las llamadas `WriteGetObjectResponse` .

Métrica	Descripción
`LambdaResponse5xx`	Es el número de errores del servidor HTTP 5xx que se produce al llamar a `WriteGetObjectResponse` desde una función de Lambda. Esta métrica proporciona la misma información que `5xxErrors` , pero solo para las llamadas `WriteGetObjectResponse` .

Métricas de Amazon S3 en Outposts en CloudWatch

Para obtener una lista de las métricas de CloudWatch que se utilizan para buckets de S3 en Outposts, consulte Métricas de CloudWatch.

Dimensiones de Amazon S3 en CloudWatch

Las siguientes dimensiones se usan para filtrar métricas de Amazon S3.

Dimensión	Descripción
`BucketName`	Esta dimensión solo filtra los datos solicitados para el bucket identificado.
`StorageType`	Esta dimensión filtra los datos que ha almacenado en un bucket mediante los siguientes tipos de almacenamiento: • `StandardStorage` : es el número de bytes que se usan en los objetos de la clase de almacenamiento `STANDARD`. • `IntelligentTieringAAStorage` : es el número de bytes que se utilizan en los objetos del nivel de acceso de archivo de la clase de almacenamiento `INTELLIGENT_TIERING` . • `IntelligentTieringAIAStorage` : es el número de bytes que se utilizan en los objetos del nivel de acceso de archivo instantáneo de la clase de almacenamiento `INTELLIGENT_TIERING` . • `IntelligentTieringDAAStorage` : es el número de bytes que se utilizan en los objetos del nivel de acceso de

Dimensión	Descripción
	archivo profundo de la clase de almacenamiento `INTELLIGENT_TIERING` . • `IntelligentTieringFAStorage` : es el número de bytes que se utilizan en los objetos del nivel de acceso frecuente de la clase de almacenamiento `INTELLIGENT_TIERING` . • `IntelligentTieringIAStorage` : es el número de bytes que se utilizan en los objetos del nivel de acceso infrecuente de la clase de almacenamiento `INTELLIGENT_TIERING` . • `StandardIAStorage` : es el número de bytes que se utiliza para los objetos de la clase S3 Standard-Infrequent Access (`STANDARD_IA`). • `StandardIASizeOverhead` : es el número de bytes que se utilizan en los objetos que tienen menos de 128 KB en la clase de almacenamiento `STANDARD_IA` . • `IntAAObjectOverhead` : para cada objeto de la clase de almacenamiento `INTELLIGENT_TIERING` en el nivel de acceso a archivos, S3 Glacier agrega 32 KB de almacenamiento para índices y metadatos relacionados. Estos datos adicionales son necesarios para identificar y restaurar su objeto. Por este almacenamiento adicional se aplican las tarifas de S3 Glacier Flexible Retrieval. • `IntAAS3ObjectOverhead` : para cada objeto de la clase de almacenamiento `INTELLIGENT_TIERING` del nivel acceso a archivos, Amazon S3 utiliza 8 KB de almacenamiento para el nombre del objeto y otros metadatos. Por este almacenamiento adicional se aplican las tarifas de S3 Standard. • `IntDAAObjectOverhead` : para cada objeto de la clase de almacenamiento `INTELLIGENT_TIERING` en el nivel de acceso a archivos profundo, S3 Glacier agrega 32 KB de almacenamiento para índices y metadatos relacionados.

Dimensión	Descripción
	Estos datos adicionales son necesarios para identificar y restaurar su objeto. Por este almacenamiento adicional se aplican las tarifas de S3 Glacier Deep Archive. • `IntDAAS3ObjectOverhead` : para cada objeto de la clase de almacenamiento `INTELLIGENT_TIERING` en el nivel de acceso a archivos profundo, Amazon S3 agrega 8 KB de almacenamiento para índices y metadatos relacionados. Estos datos adicionales son necesarios para identificar y restaurar su objeto. Por este almacenamiento adicional se aplican las tarifas de S3 Standard. • `OneZoneIAStorage` : el número de bytes que se utilizan para objetos en la clase de almacenamiento de una única zona de acceso poco frecuente de S3 (`ONEZONE_IA`). • `OneZoneIASizeOverhead` : es el número de bytes que se utilizan en los objetos que tienen menos de 128 KB en la clase de almacenamiento `ONEZONE_IA` . • `ReducedRedundancyStorage` : es el número de bytes que se utilizan en los objetos de la clase de almacenamiento de redundancia reducida (RRS). • `GlacierInstantRetrievalSizeOverhead` : es el número de bytes que se utilizan en los objetos más pequeños de 128 KB en la clase de almacenamiento S3 Glacier Instant Retrieval. • `GlacierInstantRetrievalStorage` : es el número de bytes que se utilizan en los objetos de la clase de almacenamiento S3 Glacier Instant Retrieval. • `GlacierStorage` : es el número de bytes que se utilizan en los objetos de la clase de almacenamiento S3 Glacier Flexible Retrieval. • `GlacierStagingStorage` : es el número de bytes que se utilizan para partes de los objetos multiparte antes de completar la solicitud `CompleteMultipartUpload` en

Dimensión	Descripción
	los objetos de la clase de almacenamiento S3 Glacier Flexible Retrieval. • `GlacierObjectOverhead` : por cada objeto que se archiva, S3 Glacier agrega 32 KB de almacenamiento para índices y metadatos relacionados. Estos datos adicionales son necesarios para identificar y restaurar su objeto. Por este almacenamiento adicional se aplican las tarifas de S3 Glacier Flexible Retrieval. • `GlacierS3ObjectOverhead` : por cada objeto que se archiva en S3 Glacier Flexible Retrieval, Amazon S3 utiliza 8 KB de almacenamiento para el nombre del objeto y otros metadatos. Por este almacenamiento adicional se aplican las tarifas de S3 Standard. • `DeepArchiveStorage` : es el número de bytes que se utilizan en los objetos de la clase de almacenamiento S3 Glacier Deep Archive. • `DeepArchiveObjectOverhead` : por cada objeto que se archiva, S3 Glacier agrega 32 KB de almacenamiento para índices y metadatos relacionados. Estos datos adicionales son necesarios para identificar y restaurar su objeto. Por este almacenamiento adicional se aplican las tarifas de S3 Glacier Deep Archive. • `DeepArchiveS3ObjectOverhead` : por cada objeto que se archiva en S3 Glacier Deep Archive, Amazon S3 utiliza 8 KB de almacenamiento para el nombre del objeto y otros metadatos. Por este almacenamiento adicional se aplican las tarifas de S3 Standard. • `DeepArchiveStagingStorage` : el número de bytes utilizados para partes de objetos Multipart antes de completar la solicitud `CompleteMultipartUpload` en objetos en la clase de almacenamiento S3 Glacier Deep Archive.

Dimensión	Descripción
	• `ExpressOneZoneStorage` : es el número de bytes que se usan en los objetos de la clase de almacenamiento S3 Express One Zone.
`FilterId`	Esta dimensión filtra las configuraciones de métricas especific adas para las métricas de solicitudes en un bucket. Al crear una configuración de métricas, se especifica un identificador de filtro (por ejemplo, un prefijo, una etiqueta o un punto de acceso). Para obtener más información, consulte Creating a metrics configuration (Creación de una configuración de métricas).

Dimensiones de Replicación de S3 en CloudWatch

Las siguientes dimensiones se usan para filtrar métricas de Replicación de S3.

Dimensión	Descripción
`SourceBucket`	Es el nombre de los objetos del bucket desde los que se replica.
`DestinationBucket`	Es el nombre de los objetos del bucket hacia los que se replica.
`RuleId`	Es un identificador único de la regla que ha activado la actualiza ción de esta métrica de replicación.

Dimensiones de S3 Storage Lens en CloudWatch

Para obtener una lista de las dimensiones que se utilizan para filtrar las métricas de S3 Storage Lens en CloudWatch, consulte Dimensiones.

Dimensiones de solicitud de S3 Object Lambda en CloudWatch

Las siguientes dimensiones se usan para filtrar los datos de un punto de acceso de Object Lambda.

Dimensión	Descripción
`AccessPointName`	El nombre del punto de acceso al que se realizan las solicitudes.

Dimensión	Descripción
`DataSourceARN`	Origen desde el que el punto de acceso de Object Lambda obtiene los datos. Si la solicitud llama a una función de Lambda, se refiere al Nombre de recurso de Amazon (ARN) de Lambda. De lo contrario, se refiere al ARN del punto de acceso.

Acceso a métricas de CloudWatch

Puede utilizar los siguientes procedimientos para ver las métricas de almacenamiento de Amazon S3. Para obtener las métricas de Amazon S3 pertinentes, debe establecer marcas temporales de inicio y finalización. Para las métricas correspondientes a cualquier periodo de 24 horas, establezca el periodo en 86400 segundos, el número de segundos que tiene un día. Recuerde también configurar las dimensiones `BucketName` y `StorageType`.

Uso de la AWS CLI

Por ejemplo, si quiere usar la AWS CLI para obtener el valor medio del tamaño de un bucket específico en bytes, puede usar el siguiente comando:

```
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=amzn-s3-demo-bucket
 Name=StorageType,Value=StandardStorage --period 86400 --output json
```

El ejemplo produce el siguiente resultado.

```
{
    "Datapoints": [
        {
            "Timestamp": "2016-10-19T00:00:00Z",
            "Average": 1025328.0,
            "Unit": "Bytes"
        }
    ],
    "Label": "BucketSizeBytes"
}
```

Uso de la consola de S3

Para consultar las métricas utilizando la consola de Amazon CloudWatch

1. Abra la consola de CloudWatch en https://console.aws.amazon.com/cloudwatch/.
2. En el panel de navegación izquierdo, en Metrics (Métricas).
3. Elija el espacio de nombres de S3.
4. (Opcional) Para ver una métrica, escriba el nombre de la métrica en el cuadro de búsqueda.
5. (Opcional) Para filtrar por la dimensión StorageType (Tipo de almacenamiento), escriba el nombre de la clase de almacenamiento en el cuadro de búsqueda.

Para ver una lista de métricas válidas almacenadas en su Cuenta de AWS utilizando la AWS CLI

- En el símbolo del sistema, ejecute el siguiente comando.

```
aws cloudwatch list-metrics --namespace "AWS/S3"
```

Para obtener más información acerca de los permisos necesarios para acceder a los paneles de CloudWatch, consulte Actualización de permisos del panel de CloudWatch en la Guía del usuario de Amazon CloudWatch.

Configuraciones de métricas de CloudWatch

Con las métricas de solicitudes de Amazon CloudWatch para Amazon S3, puede obtener métricas de CloudWatch de 1 minuto, configurar alarmas de CloudWatch y acceder a paneles de control de CloudWatch para ver las operaciones y el rendimiento de su almacenamiento de Amazon S3 en tiempo real. Para las aplicaciones que dependen del almacenamiento en la nube, estas métricas le permiten identificar rápidamente los problemas operativos y actuar en consecuencia. Cuando se activan, estas métricas de 1 minuto están disponibles en el nivel del bucket de Amazon S3 de forma predeterminada.

Si desea obtener las métricas de solicitudes de CloudWatch para los objetos de un bucket, debe crear una configuración de métricas para el bucket. Para obtener más información, consulte Creación de una configuración de métricas de CloudWatch para todos los objetos del bucket.

También puede usar un prefijo compartido, una etiqueta de objeto o un punto de acceso para definir las métricas recopiladas. Este método de definición de un filtro le permite crear distintos filtros de

métricas para aplicaciones empresariales, organizaciones internas o flujos de trabajo o específicos. Para obtener más información, consulte Creación de una configuración de métricas que filtra por prefijo, etiqueta de objeto o punto de acceso. Para obtener más información sobre las métricas de CloudWatch disponibles y las diferencias entre las métricas de almacenamiento y de solicitudes, consulte Monitorización de métricas con Amazon CloudWatch.

Tenga en cuenta las siguientes consideraciones al utilizar configuraciones de métricas:

- Puede tener un máximo de 1000 configuraciones de métricas por cada bucket.
- Puede seleccionar qué objetos de un bucket quiere incluir en las configuraciones de métricas mediante el uso de filtros. Puede filtrar con un prefijo compartido, una etiqueta de objeto o punto de acceso para alinear los filtros de métricas a aplicaciones empresariales, flujos de trabajo u organizaciones internas específicos. Para solicitar métricas de la totalidad del bucket, cree una configuración de métricas sin filtro.
- Las configuraciones de métricas solo son necesarias para habilitar las métricas de solicitudes. Las métricas diarias de almacenamiento en el nivel de bucket siempre están activadas y se facilitan sin costo adicional. En la actualidad no es posible obtener métricas diarias de almacenamiento para un subconjunto filtrado de objetos.
- Cada configuración de métricas habilita el conjunto completo de métricas de solicitudes disponibles. Las métricas específicas de operaciones (como `PostRequests`) solo se generan si hay solicitudes de ese tipo para el bucket o el filtro.
- Se generan métricas de solicitudes para las operaciones de objetos. También se generan para las operaciones que enumeran el contenido de un bucket, como GET Bucket (List Objects), GET Bucket Object Versions y List Multipart Uploads, pero no se generan para las demás operaciones en los buckets.
- Las métricas de solicitudes admiten el filtrado por prefijo, etiquetas de objeto o puntos de acceso, pero no las métricas de almacenamiento.

Entrega de métricas de CloudWatch en la medida de lo posible

Las métricas de CloudWatch se entregan en la medida que es posible. La mayoría de solicitudes de un objeto de Amazon S3 que tienen métricas de solicitudes se derivan en el envío de un punto de datos a CloudWatch.

La integridad y la puntualidad de las métricas no están garantizadas. Es posible que el punto de datos de una solicitud determinada se envíe con una marca temporal posterior al momento en el

que la solicitud se ha procesado realmente. Es posible que el punto de datos durante un minuto se retrase antes de estar disponible a través de CloudWatch, o puede que no se entregue en absoluto. Las métricas de solicitudes de CloudWatch le dan una idea de la naturaleza del tráfico al que se enfrenta su bucket en tiempo casi real. No pretende ser un recuento completo de todas las solicitudes.

Dado que esta característica funciona en la medida de lo posible, los informes disponibles en el panel de Administración de facturación y costos podrían incluir una o varias solicitudes de acceso que no aparecen en las métricas del bucket.

Para obtener más información sobre cómo trabajar con métricas de CloudWatch en Amazon S3, consulte los siguientes temas.

Temas

- Creación de una configuración de métricas de CloudWatch para todos los objetos del bucket
- Creación de una configuración de métricas que filtra por prefijo, etiqueta de objeto o punto de acceso
- Eliminación de un filtro de métricas

Creación de una configuración de métricas de CloudWatch para todos los objetos del bucket

Cuando configura métricas de solicitud, puede crear una configuración de métricas de CloudWatch para todos los objetos del bucket o puede filtrar por prefijo, etiqueta de objeto o punto de acceso. A través de los procedimientos de este tema, se muestra cómo crear una configuración para todos los objetos del bucket. Para crear una configuración que filtra por etiqueta de objeto, prefijo o punto de acceso, consulte Creación de una configuración de métricas que filtra por prefijo, etiqueta de objeto o punto de acceso.

Existen tres tipos de métricas de Amazon CloudWatch para Amazon S3: métricas de almacenamiento, métricas de solicitud y métricas de replicación. Las métricas de almacenamiento se informan una vez al día y se entregan a todos los clientes sin costo adicional. Las métricas de solicitudes están disponibles en intervalos de un minuto después de un breve periodo de latencia para procesarlas. Las métricas de solicitud se facturan según la tarifa de CloudWatch estándar. Debe incluir métricas de solicitudes configurándolas en la consola o con la API de Amazon S3. Las métricas de replicación de S3 proporcionan métricas detalladas para las reglas de replicación en la configuración de la misma. Con las métricas de replicación, puede monitorizar su progreso minuto a

minuto mediante el seguimiento de los bytes pendientes, las operaciones que no se han replicado y la latencia de replicación.

Para obtener más información acerca de las métricas de CloudWatch para Amazon S3, consulte Monitorización de métricas con Amazon CloudWatch.

Puede agregar configuraciones de métricas a un bucket mediante la consola de Amazon S3, AWS Command Line Interface (AWS CLI) o la API de REST de Amazon S3.

Uso de la consola de S3

1. Inicie sesión en la AWS Management Console y abra la consola de Amazon S3 en https://console.aws.amazon.com/s3/.
2. En la lista Buckets, elija el nombre del bucket que contiene los objetos para los que desea obtener métricas de solicitudes.
3. Elija la pestaña Metrics (Métricas).
4. En Bucket metrics (Métricas de bucket), seleccione View additional charts (Ver gráficos adicionales).
5. Seleccione la pestaña Request metrics (Métricas de solicitud).
6. Elija Create Filter (Crear filtro).
7. En el cuadro Filter name (Nombre del filtro), escriba el nombre del filtro.

   Los nombres solo pueden incluir letras, números, puntos, guiones y guiones bajos. Se recomienda utilizar el nombre `EntireBucket` si el filtro se aplica a todos los objetos.
8. En Filter scope (Alcance del filtro), elija This filter applies to all objects in the bucket (Este filtro se aplica a todos los objetos del bucket).

   También puede definir un filtro para que las métricas solo se recopilen y comuniquen en un subconjunto de objetos en el bucket. Para obtener más información, consulte Creación de una configuración de métricas que filtra por prefijo, etiqueta de objeto o punto de acceso.
9. Elija Save changes (Guardar cambios).
10. En la pestaña Request metrics (Métricas de solicitud), bajo Filters (Filtros), elija el filtro que acaba de crear.

    Después de unos 15 minutos, CloudWatch comienza a hacer el seguimiento de estas métricas de solicitud. Puede verlas en la pestaña Request metrics (Solicitar métricas). Puede ver gráficos

de las métricas en la consola de Amazon S3 o de CloudWatch. Las métricas de solicitud se facturan según la tarifa de CloudWatch estándar. Para obtener más información, consulte los precios de Amazon CloudWatch.

Uso de la API de REST

También puede añadir configuraciones de métricas mediante programación con la API de REST de Amazon S3. Para obtener más información acerca de cómo agregar y trabajar con configuraciones de métricas, consulte los siguientes temas en la Referencia de API de Amazon Simple Storage Service:

- PUT Bucket Metric Configuration
- GET Bucket Metric Configuration
- List Bucket Metric Configuration
- DELETE Bucket Metric Configuration

Uso de la AWS CLI

1. Instale y configure la AWS CLI. Para obtener instrucciones, consulte Instalación, actualización y desinstalación de la AWS CLI en la Guía del usuario de AWS Command Line Interface.
2. Abra un terminal.
3. Ejecute el siguiente comando para agregar una configuración de métricas.

```
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"}'
```

Creación de una configuración de métricas que filtra por prefijo, etiqueta de objeto o punto de acceso

Existen tres tipos de métricas de Amazon CloudWatch para Amazon S3: métricas de almacenamiento, métricas de solicitud y métricas de replicación. Las métricas de almacenamiento se informan una vez al día y se entregan a todos los clientes sin costo adicional. Las métricas de solicitudes están disponibles en intervalos de un minuto después de un breve periodo de latencia para procesarlas. Las métricas de solicitud se facturan según la tarifa de CloudWatch estándar. Debe incluir métricas de solicitudes configurándolas en la consola o con la API de Amazon S3. Las

métricas de replicación de S3 proporcionan métricas detalladas para las reglas de replicación en la configuración de la misma. Con las métricas de replicación, puede monitorizar su progreso minuto a minuto mediante el seguimiento de los bytes pendientes, las operaciones que no se han replicado y la latencia de replicación.

Para obtener más información acerca de las métricas de CloudWatch para Amazon S3, consulte Monitorización de métricas con Amazon CloudWatch.

Cuando configura las métricas de CloudWatch, puede crear un filtro para todos los objetos del bucket o puede filtrar la configuración en grupos de objetos relacionados dentro de un solo bucket. Puede filtrar los objetos de un bucket para incluirlos en una configuración de métricas en función de uno o varios de los siguientes tipos de filtro:

- Prefijo de nombre de clave de objeto: aunque el modelo de datos de Amazon S3 es una estructura plana, puede inferir la jerarquía con un prefijo. La consola de Amazon S3 admite estos prefijos con el concepto de carpetas. Si filtra por prefijo, los objetos que tengan el mismo prefijo se incluyen en la configuración de métricas. Para obtener más información acerca de los prefijos, consulte Organizar objetos con prefijos.
- Etiqueta: puede añadir etiquetas, que son pares de nombre de clave-valor, a los objetos. Las etiquetas le permiten encontrar y organizar los objetos fácilmente. También puede utilizar las etiquetas como filtro para las configuraciones de métricas. Para obtener más información acerca de las etiquetas de objeto, consulte Categorización del almacenamiento mediante etiquetas.
- Punto de acceso: los puntos de acceso de S3 se denominan puntos de enlace de red que se adjunta a los buckets y simplifican la administración del acceso a los datos a escala para los conjuntos de datos compartidos en S3. Cuando crea un filtro de punto de acceso, Amazon S3 incluye solicitudes al punto de acceso que especifique en la configuración de métricas. Para obtener más información, consulte Monitorización y registro de puntos de acceso.

> **Note**
>
> Al crear una configuración de métricas que filtra por punto de acceso, debe utilizar el Nombre de recurso de Amazon (ARN) del punto de acceso, no el alias del punto de acceso. Asegúrese de utilizar el ARN para el punto de acceso en sí, no el ARN para un objeto específico. Para obtener más información acerca de los ARN de puntos de acceso, consulte Usar puntos de acceso.

Si especifica un filtro, únicamente las solicitudes que operen en objetos únicos pueden coincidir con el filtro e incluirse entre las métricas de las que se informa. Las solicitudes como DeleteObjects y `ListObjects` no devuelven métricas para las configuraciones con filtros.

Para solicitar un filtrado más complejo, seleccione dos o más elementos. Solo los objetos que tengan todos estos elementos se incluirán en la configuración de métricas. Si no configura filtros, todos los objetos del bucket se incluirán en la configuración de métricas.

Uso de la consola de S3

1. Inicie sesión AWS Management Console Management Console y abra la consola de Amazon S3 en https://console.aws.amazon.com/s3/.
2. En la lista Buckets, elija el nombre del bucket que contiene los objetos para los que desea obtener métricas de solicitudes.
3. Elija la pestaña Metrics (Métricas).
4. En Bucket metrics (Métricas de bucket), seleccione View additional charts (Ver gráficos adicionales).
5. Seleccione la pestaña Request metrics (Métricas de solicitud).
6. Elija Create Filter (Crear filtro).
7. En el cuadro Filter name (Nombre del filtro), escriba el nombre del filtro.

   Los nombres solo pueden incluir letras, números, puntos, guiones y guiones bajos.
8. En Filter scope (Alcance del filtro), elija Limit the scope of this filter using a prefix, object tags, and an S3 Access Point, or a combination of all three (Limitar el alcance de este filtro usando un prefijo, etiquetas de objeto y un Punto de acceso de S3, o una combinación de las tres).
9. En Filter type (Tipo de filtro), elija al menos un tipo de filtro: Prefix (Prefijo), Object tags (Etiquetas de objeto) o Access Point (Punto de acceso).
10. Para definir un filtro de prefijo y limitar el alcance del filtro a una ruta individual, en el cuadro Prefix (Prefijo), ingrese un prefijo.
11. Para definir un filtro de etiquetas de objeto, en Object tags (Etiquetas de objeto), elija Add tag (Agregar etiqueta) y, luego, ingrese una etiqueta Key (Clave) y Value (Valor).
12. Para definir un filtro de punto de acceso, en el campo Punto de acceso de S3, introduzca el ARN del punto de acceso o elija Examinar S3 para desplazarse hasta el punto de acceso.

> **Important**
>
> No se puede introducir un alias de punto de acceso. Debe introducir el ARN para el punto de acceso en sí, no el ARN para un objeto específico.

13. Elija Guardar cambios.

    Amazon S3 crea un filtro que usa el prefijo, etiquetas o punto de acceso especificados.

14. En la pestaña Request metrics (Métricas de solicitud), bajo Filters (Filtros), elija el filtro que acaba de crear.

    Ahora ha creado un filtro que limita el alcance de las métricas de solicitud por prefijo, etiquetas de objeto o punto de acceso. Aproximadamente 15 minutos después de que CloudWatch comience a realizar el seguimiento de estas métricas de solicitudes, puede ver gráficos para las métricas en las consolas de Amazon S3 y CloudWatch. Las métricas de solicitud se facturan según la tarifa de CloudWatch estándar. Para obtener más información, consulte los precios de Amazon CloudWatch.

    También puede configurar métricas de solicitud en el nivel de bucket. Para obtener más información, consulte Creación de una configuración de métricas de CloudWatch para todos los objetos del bucket.

Uso de la AWS CLI

1. Instale y configure la AWS CLI. Para obtener instrucciones, consulte Instalación, actualización y desinstalación de la AWS CLI en la Guía del usuario de AWS Command Line Interface.
2. Abra un terminal.
3. Ejecute uno de los siguientes comandos para agregar una configuración de métricas:

   Example : para filtrar por prefijo

```
aws s3api put-bucket-metrics-configuration --bucket DOC-EXAMPLE-BUCKET1 --
id metrics-config-id --metrics-configuration '{"Id":"metrics-config-id", "Filter":
{"Prefix":"prefix1"}} '
```

Example : para filtrar por etiquetas

```
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 : para filtrar por punto de acceso

```
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 : para filtrar por prefijo, etiquetas y punto de acceso

```
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"
        }
    }
}'
```

Uso de la API de REST

También puede añadir configuraciones de métricas mediante programación con la API de REST de Amazon S3. Para obtener más información acerca de cómo agregar y trabajar con configuraciones

de métricas, consulte los siguientes temas en la Referencia de API de Amazon Simple Storage Service:

- PUT Bucket Metric Configuration
- GET Bucket Metric Configuration
- List Bucket Metric Configuration
- DELETE Bucket Metric Configuration

Eliminación de un filtro de métricas

Puede eliminar un filtro de métricas de solicitud de Amazon CloudWatch si ya no lo necesita. Al eliminar un filtro, ya no se le aplicarán cargos por las métricas de solicitud que utilicen ese filtro específico. Sin embargo, se le seguirá cobrando por cualquier otra configuración de filtro que exista.

Cuando elimina un filtro, ya no puede usar el filtro para las métricas de solicitud. La eliminación de un filtro no se puede deshacer.

Para obtener información acerca de cómo crear un filtro de métricas de solicitud, consulte los siguientes temas:

- Creación de una configuración de métricas de CloudWatch para todos los objetos del bucket
- Creación de una configuración de métricas que filtra por prefijo, etiqueta de objeto o punto de acceso

Uso de la consola de S3

1. Inicie sesión en la AWS Management Console y abra la consola de Amazon S3 en https://console.aws.amazon.com/s3/.
2. En la lista de Buckets, elija el nombre del bucket.
3. Elija la pestaña Metrics (Métricas).
4. En Bucket metrics (Métricas de bucket), seleccione View additional charts (Ver gráficos adicionales).
5. Seleccione la pestaña Request metrics (Métricas de solicitud).
6. Seleccione Manage filters (Administrar filtros).
7. Elija el filtro.

> **Important**
>
> La eliminación de un filtro no se puede deshacer.

8. Elija Delete (Eliminar).

   Amazon S3 elimina el filtro.

Uso de la API de REST

También puede añadir configuraciones de métricas mediante programación con la API de REST de Amazon S3. Para obtener más información acerca de cómo agregar y trabajar con configuraciones de métricas, consulte los siguientes temas en la Referencia de API de Amazon Simple Storage Service:

- PUT Bucket Metric Configuration
- GET Bucket Metric Configuration
- List Bucket Metric Configuration
- DELETE Bucket Metric Configuration

Notificaciones de eventos de Amazon S3

Puede utilizar la característica de notificaciones de eventos de Amazon S3 para recibir avisos cuando se produzcan ciertos eventos en el bucket de S3. Para habilitar las notificaciones, agregue una configuración de notificación que identifique los eventos que desea que Amazon S3 publique. Asegúrese de que también identifique los destinos a los que desea que Amazon S3 envíe las notificaciones. Debe almacenar esta configuración en el subrecurso de notificación asociado con un bucket. Para obtener más información, consulte Opciones de configuración de buckets. Amazon S3 proporciona una API para que administre este subrecurso.

> **Important**
>
> Las notificaciones de eventos de Amazon S3 están diseñadas para ser entregadas al menos una vez. Normalmente las notificaciones de eventos se entregan en cuestión de segundos, pero a veces pueden tardar un minuto o más.

Descripción general de las notificaciones de eventos de Amazon S3

Actualmente, Amazon S3 puede publicar notificaciones para los siguientes eventos:

- Eventos de creación de objetos nuevos
- Eventos de eliminación de objetos
- Eventos de restauración de objetos
- Eventos de pérdida de objeto de almacenamiento de redundancia reducida (RRS)
- Eventos de replicación
- Eventos de vencimiento de S3 Lifecycle
- Eventos de transición de S3 Lifecycle
- Eventos de archivo automático de S3 Intelligent-Tiering
- Eventos de etiquetado de objetos
- Eventos PUT de ACL de objetos

Para obtener una descripción completa de todos los tipos de eventos compatibles, consulte Tipos de eventos admitidos para SQS, SNS y Lambda.

Amazon S3 puede enviar mensajes de notificación de eventos a los siguientes destinos. Especifique el valor de nombre de recurso de Amazon (ARN) de estos destinos en la configuración de notificación.

- Temas de Amazon Simple Notification Service (Amazon SNS)
- Colas de Amazon Simple Queue Service (Amazon SQS)
- Función AWS Lambda
- Amazon EventBridge

Para obtener más información, consulte Destinos de eventos admitidos.

> **Note**
>
> No se admiten colas FIFO (First-In-First-Out) de Amazon Simple Queue Service como destino de la notificación de eventos de Amazon S3. Para enviar una notificación de un

evento de Amazon S3 a una cola FIFO de Amazon SQS, puede utilizar Amazon EventBridge. Para obtener más información, consulte Activación de Amazon EventBridge.

Warning

Si su notificación escribe en el bucket que desencadena la notificación, podría provocar un bucle de ejecución. Por ejemplo, si el bucket desencadena una función de Lambda cada vez que se carga un objeto y la función carga un objeto en el bucket, la función se activa indirectamente a sí misma. Para evitarlo, utilice dos buckets o configure el desencadenador para que solo se aplique a un prefijo que se utiliza para los objetos entrantes.
Para obtener más información y un ejemplo del uso de notificaciones de Amazon S3 con AWS Lambda, consulte Uso de AWS Lambda con Amazon S3 en la Guía para desarrolladores de AWS Lambda.

Para obtener más información sobre el número de configuraciones de notificación de evento que puede crear por cada bucket, consulte Cuotas de servicio de Amazon S3 en Referencia general de AWS.

Para obtener más información acerca de las notificaciones de eventos, consulte las siguientes secciones.

Temas

- Tipos y destinos de las notificaciones de eventos
- Uso de Amazon SQS, Amazon SNS y Lambda
- Uso de EventBridge

Tipos y destinos de las notificaciones de eventos

Amazon S3 admite varios tipos de notificaciones de eventos y destinos en los que se pueden publicar los avisos. Puede especificar el tipo de evento y el destino al configurar las notificaciones de eventos. Solo se puede especificar un destino para cada notificación de evento. Las notificaciones de eventos de Amazon S3 envían una entrada de evento por cada mensaje de notificación.

Temas

- Destinos de eventos admitidos

- Tipos de eventos admitidos para SQS, SNS y Lambda
- Tipos de eventos admitidos para Amazon EventBridge
- Orden de eventos y eventos duplicados

Destinos de eventos admitidos

Amazon S3 puede enviar mensajes de notificación de eventos a los siguientes destinos.

- Temas de Amazon Simple Notification Service (Amazon SNS)
- Colas de Amazon Simple Queue Service (Amazon SQS)
- AWS Lambda
- Amazon EventBridge

Sin embargo, solo se puede especificar un tipo de destino para cada notificación de evento.

> **Note**
>
> Debe conceder permisos a Amazon S3 para publicar mensajes en un tema de Amazon SNS o en una cola de Amazon SQS. También debe conceder permiso a Amazon S3 para invocar una función de AWS Lambda en su nombre. Para obtener instrucciones sobre cómo conceder estos permisos, consulte Conceder permisos para publicar mensajes de notificación de eventos en un destino.

Tema de Amazon SNS

Amazon SNS es un servicio de mensajería push flexible y totalmente administrado. Puede utilizar este servicio para enviar mensajes a dispositivos móviles o servicios distribuidos. Con SNS, puede publicar un mensaje una vez y entregarlo una o más veces. En la actualidad, el SNS estándar solo está permitido como destino de notificación de eventos S3, mientras que SNS FIFO no está permitido.

Amazon SNS coordina y administra el envío y la entrega de mensajes a los puntos de conexión o clientes que se suscriban. Puede usar la consola de Amazon S3 para crear un tema de Amazon SNS al que enviar sus notificaciones.

El tema debe estar en la misma Región de AWS que el bucket de Amazon S3. Para obtener instrucciones sobre cómo crear un tema de Amazon SNS, consulte Introducción a Amazon SNS en la Guía para desarrolladores de Amazon Simple Notification Service y las Preguntas frecuentes de Amazon SNS.

Antes de poder utilizar el tema de Amazon SNS que creó como destino de notificación de eventos, necesita lo siguiente:

- El nombre de recurso de Amazon (ARN) para el tema de Amazon SNS.
- Una suscripción válida a temas de Amazon SNS. Con ella, los suscriptores del tema reciben una notificación cuando se publica un mensaje en su tema de Amazon SNS.

Cola de Amazon SQS

Amazon SQS ofrece colas alojadas de confianza y escalables para almacenar mensajes mientras viajan entre equipos. Puede utilizar Amazon SQS para enviar cualquier volumen de datos sin la necesidad de que otros servicios tengan que estar siempre disponibles. Puede usar la consola de Amazon SQS para crear una cola de Amazon SQS a la que enviar sus notificaciones.

La cola de Amazon SQS debe estar en la misma Región de AWS que el bucket de Amazon S3. Para obtener instrucciones sobre cómo crear una cola de Amazon SQS, consulte Qué es Amazon Simple Queue Service e Introducción a Amazon SQS en la Guía para desarrolladores de Amazon Simple Queue Service.

Antes de poder usar la cola de Amazon SQS como destino de notificación de eventos, necesita lo siguiente:

- El nombre de recurso de Amazon (ARN) para la cola de Amazon SQS

> **Note**
>
> No se admiten colas FIFO (First-In-First-Out) de Amazon Simple Queue Service como destino de la notificación de eventos de Amazon S3. Para enviar una notificación de un evento de Amazon S3 a una cola FIFO de Amazon SQS, puede utilizar Amazon EventBridge. Para obtener más información, consulte Activación de Amazon EventBridge.

Lambda function

Puede utilizar AWS Lambda para ampliar otros servicios de AWS con lógica personalizada o crear su propio backend que opere con el nivel de seguridad, rendimiento y escala de AWS. Con Lambda, puede crear aplicaciones discretas basadas en eventos que se ejecutan solo cuando es necesario. También puede utilizarlo para escalar estas aplicaciones automáticamente de unas pocas solicitudes al día a miles por segundo.

Lambda puede ejecutar código personalizado en respuesta a eventos de bucket de Amazon S3. Usted carga el código personalizado a Lambda y crea lo que se llama una función de Lambda. Cuando Amazon S3 detecta un evento de un tipo específico, puede publicar el evento en AWS Lambda e invocar la función en Lambda. En respuesta, Lambda ejecuta su función. Un tipo de evento que podría detectar, por ejemplo, es un evento creado por objeto.

Puede utilizar la consola de AWS Lambda para crear una función de Lambda que utilice la infraestructura de AWS para ejecutar el código en su nombre. La función de Lambda debe estar en la misma región que el bucket de S3. También debe tener el nombre o el ARN de una función de Lambda para configurar la función de Lambda como destino de notificación de eventos.

> **Warning**
>
> Si su notificación escribe en el bucket que desencadena la notificación, podría provocar un bucle de ejecución. Por ejemplo, si el bucket desencadena una función de Lambda cada vez que se carga un objeto y la función carga un objeto en el bucket, la función se activa indirectamente a sí misma. Para evitarlo, utilice dos buckets o configure el desencadenador para que solo se aplique a un prefijo que se utiliza para los objetos entrantes.
> Para obtener más información y un ejemplo del uso de notificaciones de Amazon S3 con AWS Lambda, consulte Uso de AWS Lambda con Amazon S3 en la Guía para desarrolladores de AWS Lambda.

Amazon EventBridge

Amazon EventBridge es un bus de eventos sin servidor, que recibe eventos de servicios de AWS. Puede configurar reglas para que coincidan con los eventos y entregarlos a los objetivos, como un servicio AWS o un punto de conexión HTTP. Para obtener más información, consulte ¿Qué es EventBridge? en la Guía del usuario de Amazon EventBridge.

A diferencia de otros destinos, puede habilitar o desactivar los eventos que se entregarán a EventBridge para un bucket. Si habilita la entrega, todos los eventos se envían a EventBridge. Además, puede utilizar las reglas de EventBridge para dirigir eventos a destinos adicionales.

Tipos de eventos admitidos para SQS, SNS y Lambda

Amazon S3 puede publicar eventos de los siguientes tipos. Debe especificar estos tipos de eventos en la configuración de notificación.

Tipos de eventos	Descripción
S3:TestEvent	Cuando se habilita una notificación, Amazon S3 publica una notificación de prueba. Con esto, se busca garantizar que el tema existe y que el propietario del bucket tiene permiso para publicar el tema especificado. Si la activación de la notificación falla, no recibirá una notificación de prueba.
s3:ObjectCreated:* s3:ObjectCreated:Put s3:ObjectCreated:Post s3:ObjectCreated:Copy s3:ObjectCreated:CompleteMultipartUpload	Las operaciones de API de Amazon S3 como PUT, POST y COPY pueden crear un objeto. Con estos tipos de eventos, puede habilitar las notificaciones cuando se crea un objeto mediante una operación de API específica. También puede utilizar el tipo de evento `s3:ObjectCreated:*` para solicitar una notificación independientemente de la API que se utilizó para crear un objeto. `s3:ObjectCreated:CompleteMultipartUpload` incluye objetos que se crean con UploadPartCopy para operaciones de copia.
s3:ObjectRemoved:* s3:ObjectRemoved:Delete s3:ObjectRemoved:DeleteMarkerCreated	Con los tipos de eventos ObjectRemoved, puede habilitar la notificación cuando se elimina un objeto o un lote de objetos de un bucket. Puede solicitar una notificación cuando un objeto se elimina o un objeto con control de versiones se elimina de forma permanente con el tipo de evento `s3:ObjectRemoved:Delete` . También puede solicitar una notificación cuando

Tipos de eventos	Descripción
	se crea un marcador de eliminación para un objeto con control de versiones con `s3:ObjectRemoved:DeleteMarkerCreated`. Para obtener instrucciones sobre cómo eliminar objetos con control de versiones, consulte Eliminar versiones de objetos de un bucket con control de versiones habilitado. También puede utilizar un comodín `s3:ObjectRemoved:*` para solicitar una notificación cada vez que se elimina un objeto. Estas notificaciones de eventos no lo alertan sobre eliminaciones automáticas de configuraciones del ciclo de vida o de operaciones fallidas.
s3:ObjectRestore:* s3:ObjectRestore:Post s3:ObjectRestore:Completed s3:ObjectRestore:Delete	Con los tipos de eventos ObjectRestore, puede recibir notificaciones de iniciación y finalización de eventos al restaurar objetos desde la clase de almacenamiento S3 Glacier Flexible Retrieval, la clase de almacenamiento S3 Glacier Deep Archive, el nivel S3 Intelligent-Tierin g Archive Access y el nivel S3 Intelligent-Tiering Deep Archive Access. También puede recibir notificaciones de cuándo vence la copia restaurada de un objeto. El tipo de evento `s3:ObjectRestore:Post` le notifica sobre el inicio de la restauración de los objetos. El tipo de evento `s3:ObjectRestore:Completed` le notifica sobre la finalización de la restauración. El tipo de evento `s3:ObjectRestore:Delete` le notifica cuando vence la copia temporal de un objeto restaurado.
s3:ReducedRedundancyLostObject	Puede recibir este evento de notificación cuando Amazon S3 detecte que se ha perdido un objeto de la clase de almacenamiento RRS.

Tipos de eventos	Descripción
s3:Replication:* s3:Replication:OperationFailedReplication s3:Replication:OperationMissedThreshold s3:Replication:OperationReplicatedAfterThreshold s3:Replication:OperationNotTracked	Mediante el uso de tipos de eventos de replicación, puede recibir notificaciones para configuraciones de replicación que indican que se han habilitado las métricas de replicación de S3 o el control de tiempo de replicación de S3 (S3 RTC). Puede monitorear el progreso de los eventos de replicación minuto a minuto mediante el seguimiento de los bytes y las operaciones pendientes, y la latencia de replicación. Para obtener información acerca de las métricas de replicación, consulte Monitoreo del progreso con métricas de replicación y notificaciones de eventos de S3. El tipo de evento `s3:Replication:OperationFailedReplication` le notifica cuando un objeto que era apto para la replicación no pudo replicarse. El tipo de evento `s3:Replication:OperationMissedThreshold` le notifica cuando un objeto que era apto para la replicación supera el umbral de 15 minutos para la replicación. El tipo de evento `s3:Replication:OperationReplicatedAfterThreshold` le notifica cuando un objeto que era apto para la replicación mediante S3 Replication Time Control (S3 RTC, Control del tiempo de replicación de S3) se replica después del umbral de 15 minutos. El tipo de evento `s3:Replication:OperationNotTracked` le notifica cuando un objeto que era apto para la replicación mediante S3 Replication Time Control (S3 RTC, Control del tiempo de replicación de S3) deja de someterse al seguimiento de las métricas de replicación.

Tipos de eventos	Descripción
s3:LifecycleExpiration:* s3:LifecycleExpiration:Delete s3:LifecycleExpiration:DeleteMarkerCreated	Mediante el uso de los tipos de eventos LifecycleExpiration, puede recibir una notificación cuando Amazon S3 elimina un objeto según la configuración de S3 Lifecycle. El tipo de evento `s3:LifecycleExpiration:Delete` le notifica cuando se elimina un objeto de un bucket sin control de versiones. También le notifica cuando la versión de un objeto se elimina de forma permanente mediante una configuración de S3 Lifecycle. El tipo de evento `s3:LifecycleExpiration:DeleteMarkerCreated` le notifica cuando S3 Lifecycle crea un marcador de eliminación cuando se elimina la versión actual de un objeto del bucket con control de versiones.
s3:LifecycleTransition	Recibirá este evento de notificación cuando un objeto se transfiera a otra clase de almacenamiento de Amazon S3 mediante una configuración de S3 Lifecycle.
s3:IntelligentTiering	Recibirá este evento de notificación cuando un objeto de la clase de almacenamiento S3 Intelligent-Tiering se transfier a al nivel Archive Access o Deep Archive Access.
s3:ObjectTagging:* s3:ObjectTagging:Put s3:ObjectTagging:Delete	Mediante el uso de los tipos de eventos ObjectTagging, puede habilitar la notificación cuando se agrega o elimina la etiqueta de un objeto de un bucket. El tipo de evento `s3:ObjectTagging:Put` le notifica cuando una etiqueta es PUT en un objeto o se actualiza una etiqueta existente. El tipo de evento `s3:ObjectTagging:Delete` le notifica cuando se quita una etiqueta de un objeto.
s3:ObjectAcl:Put	Recibe este evento de notificación cuando una ACL es PUT en un objeto o cuando se cambia una ACL existente. No se genera un evento cuando una solicitud no produce cambios en la ACL de un objeto.

Tipos de eventos admitidos para Amazon EventBridge

Para obtener una lista de los tipos de eventos que Amazon S3 enviará a Amazon EventBridge, consulte Uso de EventBridge.

Orden de eventos y eventos duplicados

Las notificaciones de eventos de Amazon S3 se han diseñado para entregar notificaciones al menos una vez, pero no se garantiza que lleguen en el mismo orden en que se produjeron los eventos. En casos excepcionales, el mecanismo de reintento de Amazon S3 puede originar notificaciones de eventos de S3 duplicadas para el mismo evento de objeto. Para obtener más información sobre la gestión de eventos duplicados o fuera de servicio, consulte Administrar los pedidos de eventos y los eventos duplicados con las notificaciones de eventos de Amazon S3 en el Blog de almacenamiento de AWS.

Uso de Amazon SQS, Amazon SNS y Lambda

La habilitación de notificaciones es una operación de bucket. Almacene la información de configuración de notificaciones en el subrecurso de notificación asociado a un bucket. Después de crear o cambiar la configuración de notificación del bucket, normalmente lleva alrededor de cinco minutos para que los cambios surtan efecto. Cuando la notificación se habilita por primera vez, se produce `s3:TestEvent`. Puede utilizar cualquiera de los siguientes métodos para administrar la configuración de notificación:

- Uso de la consola de Amazon S3: puede utilizar la interfaz de usuario de la consola para configurar una notificación en un bucket sin tener que escribir ningún código. Para obtener más información, consulte Habilitación y configuración de notificaciones de eventos mediante la consola de Amazon S3.
- Uso de los SDK de AWS mediante programación: internamente, tanto la consola como los SDK llaman a la API de REST de Amazon S3 para administrar los subrecursos de notificación asociados al bucket. Para obtener ejemplos de configuraciones de notificación que utilizan AWS SDK, consulte Explicación: configuración de un bucket para notificaciones (tema de SNS o cola de SQS).

> **Note**
>
> También puede realizar llamadas a la API de REST de Amazon S3 directamente desde su código. Sin embargo, esto puede ser engorroso porque para ello debe escribir el código para autenticar las solicitudes.

Independientemente del método que utilice, Amazon S3 almacena la configuración de notificación como XML en el subrecurso de notificación asociado a un bucket. Para obtener información acerca de los subrecursos de bucket, consulte Opciones de configuración de buckets.

Temas

- Conceder permisos para publicar mensajes de notificación de eventos en un destino
- Habilitación y configuración de notificaciones de eventos mediante la consola de Amazon S3
- Configuración de notificaciones de eventos mediante programación
- Explicación: configuración de un bucket para notificaciones (tema de SNS o cola de SQS)
- Configuración de notificaciones de eventos mediante el filtrado de nombres de clave de objeto
- Estructura de mensaje de evento

Conceder permisos para publicar mensajes de notificación de eventos en un destino

Debe conceder a la entidad principal de Amazon S3 los permisos necesarios para llamar a la API pertinente a fin de publicar mensajes en un tema SNS, una cola SQS o una función de Lambda. Esto permite que Amazon S3 pueda publicar mensajes de notificación de eventos en un destino.

Para solucionar problemas de publicación de mensajes de notificación de eventos en un destino, consulte Solucionar problemas para publicar notificaciones de eventos de Amazon S3 en un tema de Amazon Simple Notification Service.

Temas

- Concesión de permisos para invocar una función de AWS Lambda
- Conceder permisos para publicar mensajes en un tema de SNS o una cola de SQS

Concesión de permisos para invocar una función de AWS Lambda

Para publicar mensajes de eventos en AWS Lambda, Amazon S3 invoca una función de Lambda y proporciona el mensaje de evento como un argumento.

Cuando utiliza la consola de Amazon S3 para configurar notificaciones de eventos en un bucket de Amazon S3 para una función de Lambda, la consola configura los permisos necesarios en la función de Lambda. Esto permite que Amazon S3 tenga permisos para invocar la función desde el bucket. Para obtener más información, consulte Habilitación y configuración de notificaciones de eventos mediante la consola de Amazon S3.

También puede conceder permisos a Amazon S3 desde AWS Lambda para invocar su función de Lambda. Para obtener más información, consulte Tutorial: utilizar AWS Lambda con Amazon S3 en la Guía para desarrolladores de AWS Lambda.

Conceder permisos para publicar mensajes en un tema de SNS o una cola de SQS

Debe asociar una política de AWS Identity and Access Management (IAM) al tema de SNS o la cola de SQS de destino a fin de conceder a Amazon S3 permisos para publicar mensajes en el tema de SNS o la cola de SQS.

Para ver un ejemplo de cómo asociar una política a un tema de SNS o una cola de SQS, consulte Explicación: configuración de un bucket para notificaciones (tema de SNS o cola de SQS). Para obtener más información acerca de estos permisos, consulte los siguientes temas:

- Casos de ejemplo para el control de acceso de Amazon SNS en la Guía para desarrolladores de Amazon Simple Notification Service
- Identity and Access Management en Amazon SQS en la guía para desarrolladores de Amazon Simple Queue Service

Política de IAM para un tema de SNS de destino

A continuación, se muestra un ejemplo de una política de AWS Identity and Access Management (IAM) que se asocia al tema de SNS de destino. Para obtener instrucciones sobre cómo utilizar esta política para configurar un tema de Amazon SNS de destino para las notificaciones de eventos, consulte Explicación: configuración de un bucket para notificaciones (tema de SNS o cola de 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"
                }
            }
        }
    ]
}
```

Política de IAM para una cola de SQS de destino

A continuación se muestra un ejemplo de una política de IAM que se asocia a la cola de SQS de destino. Para obtener instrucciones sobre cómo utilizar esta política para configurar una cola de Amazon SQS de destino para las notificaciones de eventos, consulte Explicación: configuración de un bucket para notificaciones (tema de SNS o cola de SQS).

Para utilizar esta política, debe actualizar el ARN de la cola de Amazon SQS, el nombre del bucket y el ID de la Cuenta de AWS del propietario del 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"
                }
            }
        }
    ]
}
```

Tanto para las políticas de IAM de Amazon SNS como de Amazon SQS puede especificar la condición `StringLike` en la política, en lugar de la condición `ArnLike`.

Cuando se usa `ArnLike`, las partes del ARN de partición, servicio, ID de cuenta, tipo de recurso e ID de recurso parcial deben coincidir exactamente con el ARN en el contexto de la solicitud. Solo la región y la ruta del recurso permiten la coincidencia parcial.

Cuando se usa `StringLike` en lugar de `ArnLike`, la coincidencia ignora la estructura del ARN y permite la coincidencia parcial, independientemente de la parte marcada como comodín. Para obtener más información, consulte Elementos de la política JSON de IAM en la Guía del usuario de IAM.

```
"Condition": {
  "StringLike": { "aws:SourceArn": "arn:aws:s3:*:*:bucket-name" }
  }
```

Política de claves de AWS KMS

Si la cola de SQS o los temas de SNS están cifrados con una clave administrada por el cliente AWS Key Management Service (AWS KMS), debe conceder a la entidad principal del servicio Amazon S3 permiso para trabajar con la cola o los temas cifrados. Para conceder el permiso principal de servicio de Amazon S3, agregue la siguiente declaración a la política de claves para la clave administrada por el cliente.

```
{
```

```
    "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": "*"
        }
    ]
}
```

Para obtener más información acerca de las políticas de claves de AWS KMS, consulte Uso de las políticas de claves en AWS KMS en la Guía para desarrolladores de AWS Key Management Service.

Para obtener más información acerca del uso del cifrado del lado del servidor con AWS KMS para Amazon SQS y Amazon SNS, consulte lo siguiente:

- Key management (Administración de claves) en la Guía para desarrolladores de Amazon Simple Notification Service.
- Key management (Administración de claves) en la Guía para desarrolladores de Amazon Simple Queue Service.
- Encrypting messages published to Amazon SNS with AWS KMS en el Blog de cómputo de AWS.

Habilitación y configuración de notificaciones de eventos mediante la consola de Amazon S3

Puede habilitar determinados eventos de bucket de Amazon S3 para enviar un mensaje de notificación a un destino cada vez que se produzcan estos eventos. En esta sección se explica cómo usar la consola de Amazon S3 para habilitar la notificación de eventos. Para obtener información acerca de cómo utilizar notificaciones de eventos con los SDK de AWS y las API de REST de Amazon S3, consulte Configuración de notificaciones de eventos mediante programación.

Requisitos previos: antes de habilitar las notificaciones de eventos para el bucket, debe configurar uno de los tipos de destino y, a continuación, configurar los permisos. Para obtener más información, consulte Destinos de eventos admitidos y Conceder permisos para publicar mensajes de notificación de eventos en un destino.

> **Note**
>
> No se admiten colas FIFO (First-In-First-Out) de Amazon Simple Queue Service como destino de la notificación de eventos de Amazon S3. Para enviar una notificación de un evento de Amazon S3 a una cola FIFO de Amazon SQS, puede utilizar Amazon EventBridge. Para obtener más información, consulte Activación de Amazon EventBridge.

Temas

- Habilitación de notificaciones de Amazon SNS, Amazon SQS o Lambda mediante la consola de Amazon S3

Habilitación de notificaciones de Amazon SNS, Amazon SQS o Lambda mediante la consola de Amazon S3

Para habilitar y configurar notificaciones de eventos para un bucket de S3

1. Inicie sesión en la AWS Management Console y abra la consola de Amazon S3 en https://console.aws.amazon.com/s3/.
2. En la lista Buckets, seleccione el nombre del bucket para el que desea habilitar eventos.
3. Seleccione Properties (Propiedades).
4. Acceda a la sección Notificaciones de eventos y elija Creación de notificación de eventos.
5. En la sección Configuración general, especifique el nombre del evento descriptivo para la notificación de eventos. Opcionalmente, también puede especificar un prefijo y un sufijo para limitar las notificaciones a objetos con claves que terminen en los caracteres especificados.

    a. Introduzca una descripción para el nombre del evento.

    Si no escribe un nombre, se genera un identificador único global (GUID) que se utiliza para el nombre.

    b. (Opcional) Para filtrar notificaciones de eventos por prefijo, ingrese un prefijo.

Por ejemplo, puede configurar un filtro de prefijo para recibir notificaciones solo cuando se añadan archivos a una carpeta específica (por ejemplo, `images/`).

c. (Opcional) Para filtrar notificaciones de eventos por sufijo, ingrese un sufijo.

Para obtener más información, consulte Configuración de notificaciones de eventos mediante el filtrado de nombres de clave de objeto.

6. En la sección Event types (Tipos de evento), seleccione uno o varios tipos de eventos para los que desee recibir notificaciones.

Para obtener una lista de los diferentes tipos de eventos, consulte Tipos de eventos admitidos para SQS, SNS y Lambda.

7. En la sección Destino, elija el destino de notificación de eventos.

> **Note**
>
> Antes de poder publicar notificaciones de eventos, debe conceder a la entidad principal de Amazon S3 los permisos necesarios para llamar a la API correspondiente. De este modo, puede publicar notificaciones en una función de Lambda, un tema SNS o una cola SQS.

a. Seleccione el tipo de destino: función de Lambda, Tema SNS o Cola SQS.

b. Después de elegir el tipo de destino, elija una función, un tema o una cola de la lista.

c. O bien, si prefiere especificar un nombre de recurso de Amazon (ARN), seleccione Enter ARN (Escribir ARN) y escríbalo.

Para obtener más información, consulte Destinos de eventos admitidos.

8. Seleccione Save changes (Guardar cambios)y Amazon S3 enviará un mensaje de prueba al destino de notificación de eventos.

Configuración de notificaciones de eventos mediante programación

De forma predeterminada, las notificaciones no están habilitadas para ningún tipo de evento. Por lo tanto, el subrecurso de notificación inicialmente almacena las configuraciones vacías.

```
<NotificationConfiguration xmlns="http://s3.amazonaws.com/doc/2006-03-01/">
</NotificationConfiguration>
```

Para permitir las notificaciones de eventos de determinado tipo, debe reemplazar el XML con la configuración adecuada que identifique los tipos de eventos que desea que Amazon S3 publique y el destino donde desea publicar los eventos. Para cada destino, debe añadir la configuración XML correspondiente.

Para publicar mensajes de evento en una cola SQS

A fin de establecer una cola SQS como destino de notificación para uno o varios tipos de eventos, agregue `QueueConfiguration`.

```
<NotificationConfiguration>
  <QueueConfiguration>
    <Id>optional-id-string</Id>
    <Queue>sqs-queue-arn</Queue>
    <Event>event-type</Event>
    <Event>event-type</Event>
     ...
  </QueueConfiguration>
   ...
</NotificationConfiguration>
```

Para publicar mensajes de evento en un tema SNS

A fin de establecer un tema SNS como destino de notificación para tipos de eventos específicos, agregue `TopicConfiguration`.

```
<NotificationConfiguration>
  <TopicConfiguration>
     <Id>optional-id-string</Id>
     <Topic>sns-topic-arn</Topic>
     <Event>event-type</Event>
     <Event>event-type</Event>
      ...
  </TopicConfiguration>
   ...
</NotificationConfiguration>
```

Para invocar la función de AWS Lambda y proporcionar un mensaje de evento como argumento

A fin de establecer una función de Lambda como destino de notificación para tipos de eventos específicos, agregue `CloudFunctionConfiguration`.

```
<NotificationConfiguration>
  <CloudFunctionConfiguration>
     <Id>optional-id-string</Id>
     <CloudFunction>cloud-function-arn</CloudFunction>
     <Event>event-type</Event>
     <Event>event-type</Event>
      ...
  </CloudFunctionConfiguration>
   ...
</NotificationConfiguration>
```

Para eliminar todas las notificaciones configuradas en un bucket

Para eliminar todas las notificaciones configuradas en un bucket, guarde un `<NotificationConfiguration/>` elemento vacío en el subrecurso de notificación .

Cuando Amazon S3 detecta un evento del tipo específico, publica un mensaje con la información del evento. Para obtener más información, consulte Estructura de mensaje de evento.

Para obtener más información acerca de la configuración de notificaciones de eventos, consulte los temas siguientes:

- Explicación: configuración de un bucket para notificaciones (tema de SNS o cola de SQS).
- Configuración de notificaciones de eventos mediante el filtrado de nombres de clave de objeto

Explicación: configuración de un bucket para notificaciones (tema de SNS o cola de SQS)

Puede recibir notificaciones de Amazon S3 mediante Amazon Simple Notification Service (Amazon SNS) o Amazon Simple Queue Service (Amazon SQS). En este tutorial, agregue una configuración de notificación al bucket mediante un tema de Amazon SNS y una cola de Amazon SQS.

> **Note**
>
> No se admiten colas FIFO (First-In-First-Out) de Amazon Simple Queue Service como destino de la notificación de eventos de Amazon S3. Para enviar una notificación de un

evento de Amazon S3 a una cola FIFO de Amazon SQS, puede utilizar Amazon EventBridge. Para obtener más información, consulte Activación de Amazon EventBridge.

Temas

Resumen del tutorial

Esta explicación lo ayuda a hacer lo siguiente:

- Publique eventos de tipo `s3:ObjectCreated:*` en una cola de Amazon SQS.
- que publique eventos de tipo `s3:ReducedRedundancyLostObject` en un tema de Amazon SNS.

Para obtener información acerca de la configuración de notificación, consulte Uso de Amazon SQS, Amazon SNS y Lambda.

Puede realizar todos estos pasos con la consola sin escribir ningún código. Además, se brindan ejemplos de código con los AWS SDK para Java y .NET, con el fin de ayudarlo a agregar configuraciones de notificación mediante programación.

El procedimiento incluye los pasos siguientes:

1. Cree una cola de Amazon SQS.

   Con la consola de Amazon SQS, cree una cola de SQS. Puede acceder a cualquier mensaje que Amazon S3 envía a la cola mediante programación. Sin embargo, para esta explicación, va a verificar los mensajes de notificación en la consola.

   Debe asociar una política de acceso a la cola para otorgarle permiso a Amazon S3 para publicar mensajes.
2. Cree un tema de Amazon SNS.

Con la consola de Amazon SNS, cree un tema SNS y suscríbase al tema. De esta forma, cualquier evento que se publique en él se le entregará a usted. Debe especificar el correo electrónico como protocolo de comunicación. Después de crear un tema, Amazon SNS envía un correo electrónico. Utilice el enlace del correo electrónico para confirmar la suscripción al tema.

Debe asociar una política de acceso al tema para otorgarle a Amazon S3 permiso para publicar mensajes.

3. Añada una configuración de notificación a un bucket.

Paso 1: Crear una cola de Amazon SQS

Siga los pasos para crear y suscribirse a una cola de Amazon Simple Queue Service (Amazon SQS)

1. Con la consola de Amazon SQS, cree una cola. Para obtener instrucciones, consulte Getting Started with Amazon SQS en la Amazon Simple Queue Service Developer Guide.
2. Sustituya la política de acceso asociada a la cola por la siguiente política.

   a. En la consola de Amazon SQS, en la lista de colas, elija el nombre de la cola.

   b. En la pestaña Access policy (Política de acceso), elija Edit (Editar).

   c. Sustituya la política de acceso asociada a la cola. En ella, proporcione el ARN de Amazon SQS, el nombre del bucket de origen y el ID de cuenta del propietario del bucket.

```
{
    "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. Seleccione Guardar.

3. (Opcional) Si la cola de Amazon SQS o el tema de Amazon SNS tienen habilitado el cifrado del lado del servidor con AWS Key Management Service (AWS KMS), agregue la siguiente política a la clave administrada por el cliente de cifrado simétrica.

   Debe agregar la política a una clave administrada por el cliente porque no puede modificar la clave administrada por AWS para Amazon SQS o 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": "*"
        }
    ]
}
```

   Para obtener más información acerca del uso de SSE para Amazon SQS y Amazon SNS con AWS KMS, consulte lo siguiente:

   - Key management (Administración de claves) en la Guía para desarrolladores de Amazon Simple Notification Service.

- Key management (Administración de claves) en la Guía para desarrolladores de Amazon Simple Queue Service.

4. Anote el ARN de la cola.

   La cola SQS que creó es otro recurso de la Cuenta de AWS. Tiene un nombre de recurso de Amazon (ARN) único. Necesitará este ARN en el siguiente paso. El ARN tiene el siguiente formato:

   ```
   arn:aws:sqs:aws-region:account-id:queue-name
   ```

Paso 2: Crear un tema sobre Amazon SNS

Siga los pasos para crear y suscribirse a un tema de Amazon SNS.

1. Con la consola de Amazon SNS, cree un tema. Para obtener instrucciones, consulte el tema Creación de un Amazon SNS en la Guía para desarrolladores de Amazon Simple Notification Service.
2. Suscríbase al tema. Para este ejercicio, utilice el correo electrónico como el protocolo de comunicación. Para obtener instrucciones, consulte el tema Suscribirse a un Amazon SNS en la Guía para desarrolladores de Amazon Simple Notification Service.

   Recibirá un correo electrónico donde deberá confirmar su suscripción al tema. Confirme la suscripción.
3. Sustituya la política de acceso asociada al tema por la siguiente política. En ella, proporcione el ARN del tema de SNS, el nombre del bucket y el ID de cuenta del propietario del bucket.

   ```
   {
       "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. Apunte el ARN del tema.

   El tema de SNS que creó es otro recurso de su Cuenta de AWS y tiene un ARN único. Necesitará este ARN en el siguiente paso. El ARN tendrá el siguiente formato:

   ```
   arn:aws:sns:aws-region:account-id:topic-name
   ```

Paso 3: Agregar una configuración de notificación al bucket

Puede habilitar las notificaciones del bucket con la consola de Amazon S3 o mediante programación con los AWS SDK. Seleccione cualquiera de las opciones para configurar las notificaciones en su bucket. En esta sección se brindan ejemplos de códigos en los que se utilizan los AWS SDK para Java y .NET.

Opción A: habilitar notificaciones en un bucket mediante la consola

Mediante la consola de Amazon S3, agregue una configuración de notificación al solicitar a Amazon S3 que realice lo siguiente:

- Publicar eventos del tipo Todos los objeto crean eventos en su cola de Amazon SQS.
- Publicar eventos del tipo Objeto perdido en RRS para su tema de Amazon SNS.

Después de guardar la configuración de notificación, Amazon S3 publica un mensaje de prueba que usted recibe por correo electrónico.

Para obtener instrucciones, consulte Habilitación y configuración de notificaciones de eventos mediante la consola de Amazon S3.

Opción B: habilitar notificaciones en un bucket mediante los AWS SDK

.NET

El siguiente ejemplo de código C# brinda un listado completo de códigos que añade una configuración de notificación a un bucket. Debe actualizar el código y proporcionar el nombre del bucket y el ARN del tema de SNS. Para obtener información acerca de cómo configurar y ejecutar ejemplos de código, consulte Introducción al SDK de AWS para .NET en la Guía para desarrolladores del SDK de AWS para .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

El siguiente ejemplo muestra cómo añadir una configuración de notificación a un bucket. Para obtener instrucciones sobre cómo crear y probar una muestra funcional, consulte Introducción en la Guía del desarrollador de AWS SDK for Java.

```
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();
        }
    }
}
```

Paso 4: Probar la configuración

Ahora puede cargar un objeto en el bucket y verificar la notificación de eventos en la consola de Amazon SQS para probar la configuración. Para obtener instrucciones, consulte Receiving a Message en la sección "Getting Started" de Amazon Simple Queue Service Developer Guide.

Configuración de notificaciones de eventos mediante el filtrado de nombres de clave de objeto

Al configurar una notificación de evento de Amazon S3, debe especificar qué tipos de eventos de Amazon S3 admitidos hacen que Amazon S3 envíe la notificación. Si se produce un tipo de evento que no ha especificado en el bucket de S3, Amazon S3 no envía la notificación.

Puede configurar las notificaciones para que se filtren por el prefijo y el sufijo del nombre de clave de objetos. Por ejemplo, puede establecer una configuración para recibir una notificación solo cuando se agregan archivos de imagen con una extensión de nombre de archivo "`.jpg`" a un bucket. O bien, puede establecer una configuración que envíe una notificación a un tema de Amazon SNS cuando se agregue un objeto con el prefijo "`images/`" al bucket y, al mismo tiempo, enviar notificaciones por objetos con un prefijo "`logs/`" en el mismo bucket a una función de AWS Lambda.

> **Note**
>
> No se puede usar un carácter comodín ("*") en los filtros como prefijo o sufijo. Si el prefijo o el sufijo contienen un espacio, debe sustituirlo por el carácter "+". Si utiliza otros caracteres especiales en el valor del prefijo o sufijo, debe escribirlos en formato codificado en URL (codificado en porcentaje). Para obtener una lista completa de los caracteres especiales que se deben convertir a un formato codificado en URL cuando se utilizan en un prefijo o sufijo para las notificaciones de eventos, consulte Caracteres seguros.

Puede establecer configuraciones de notificaciones que utilicen el filtrado de nombre de clave de objeto en la consola de Amazon S3. Para ello, puede utilizar las API de Amazon S3 a través de los SDK de AWS o las API de REST directamente. Para obtener información sobre el uso de la interfaz de usuario de la consola para establecer una configuración de notificación en un bucket, consulte Habilitación y configuración de notificaciones de eventos mediante la consola de Amazon S3.

Amazon S3 guarda la configuración de notificación como XML en el subrecurso notificación asociado a un bucket según lo descrito en Uso de Amazon SQS, Amazon SNS y Lambda. Puede utilizar la estructura XML `Filter` para definir las reglas para filtrar las notificaciones según el prefijo o sufijo de un nombre de clave de objeto. Para obtener información acerca de la estructura XML de `Filter`, consulte Notificación de bucket PUT en la Referencia de la API de Amazon Simple Storage Service.

Las configuraciones de notificaciones que utilizan `Filter` no pueden definir las reglas de filtrado con prefijos superpuestos, sufijos superpuestos o superposición de prefijo y sufijo. En las siguientes secciones se presentan ejemplos de configuraciones de notificación válidas con filtrado de nombres de clave de objeto. También contienen ejemplos de configuraciones de notificación que no son válidas debido a la superposición de prefijo y sufijo.

Temas

- Ejemplos de configuraciones de notificaciones válidas con filtrado de nombre de clave de objeto
- Ejemplos de configuraciones de notificación con superposición de prefijo o sufijo no válido

Ejemplos de configuraciones de notificaciones válidas con filtrado de nombre de clave de objeto

La siguiente configuración de notificación incluye una configuración de cola que identifica una cola de Amazon SQS donde Amazon S3 puede publicar eventos del tipo `s3:ObjectCreated:Put`. Los eventos se publican siempre que se envíe una solicitud PUT para un objeto con un prefijo de `images/` y un sufijo `jpg` a un bucket.

```
<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 siguiente configuración de notificación tiene varios prefijos no superpuestos. La configuración define que las notificaciones para solicitudes PUT en la carpeta `images/` van a la cola A, mientas que las notificaciones para solicitudes PUT en la carpeta `logs/` van a la cola 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 siguiente configuración de notificación tiene varios sufijos no superpuestos. Mediante la configuración, se define que todas las imágenes `.jpg` recientemente agregadas al bucket se procesan mediante la función de nube A de Lambda y todas las imágenes `.png` recientemente agregadas se procesan mediante la función de nube B. Los sufijos `.png` y `.jpg` no se superponen a pesar de terminar con la misma letra. Si una determinada cadena puede finalizar con ambos sufijos, los dos sufijos se consideran superpuestos. Una cadena no puede finalizar con `.png` y `.jpg`, por lo que los sufijos en la configuración de ejemplo no son sufijos superpuestos.

```
<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>
```

Las configuraciones de notificación que utilizan `Filter` no pueden definir reglas de filtrado con prefijos superpuestos para los mismos tipos de eventos. Solo pueden hacerlo si los prefijos superpuestos que se utilizan con sufijos que no se superponen. La siguiente configuración de

ejemplo muestra cómo los objetos creados con un prefijo común pero sufijos no superpuestos se pueden enviar a diferentes destinos.

```
<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>
```

Ejemplos de configuraciones de notificación con superposición de prefijo o sufijo no válido

En su mayoría, las configuraciones de notificación que utilizan `Filter` no pueden definir reglas de filtrado con prefijos superpuestos, sufijos superpuestos ni combinaciones de prefijos y sufijos superpuestos para los mismos tipos de eventos. Puede tener prefijos superpuestos siempre que los sufijos no se superpongan. Para ver un ejemplo, consulte Configuración de notificaciones de eventos mediante el filtrado de nombres de clave de objeto.

Puede utilizar filtros de nombre de clave de objeto superpuestos con diferentes tipos de eventos. Por ejemplo, puede crear una configuración de notificación que utilice el prefijo `image/` para el tipo de evento `ObjectCreated:Put` y el prefijo `image/` para el tipo de evento `ObjectRemoved:*`.

Se recibe un mensaje de error si se intenta guardar una configuración de notificación que tiene filtros de nombre superpuestos no válidos para los mismos tipos de eventos, o cuando se utiliza la consola o la API de Amazon S3. En esta sección, se muestran ejemplos de configuraciones de notificación que no son válidas debido a los filtros de nombre superpuestos.

Se asume que cualquier regla de configuración de notificación existente tiene un prefijo y un sufijo predeterminados que coinciden con cualquier otro prefijo y sufijo respectivamente. La siguiente configuración de notificación no es válida porque incluye prefijos superpuestos. En concreto, el prefijo raíz se superpone con cualquier otro prefijo. Lo mismo sucede si utiliza un sufijo en lugar de un prefijo en este ejemplo. El sufijo raíz se superpone con cualquier otro sufijo.

```
<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 siguiente configuración de notificación no es válida porque incluye sufijos superpuestos. Si una determinada cadena puede finalizar con ambos sufijos, los dos sufijos se consideran superpuestos. Una cadena puede finalizar con `jpg` y `pg`. Por lo tanto, los sufijos se superponen. Lo mismo se aplica a los prefijos. Si una cadena dada puede comenzar con ambos prefijos, los dos prefijos se consideran superpuestos.

```
 <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 siguiente configuración de notificación no es válida porque incluye prefijos y sufijos superpuestos.

```
<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>
```

Estructura de mensaje de evento

El mensaje de notificación que Amazon S3 envía para publicar un evento está en formato JSON.

Para obtener información general e instrucciones sobre cómo configurar notificaciones de eventos, consulte Notificaciones de eventos de Amazon S3.

En este ejemplo se muestra versión 2.2 de la estructura JSON de notificación de eventos. Amazon S3 utiliza versiones 2.1, 2.2 y 2.3 de esta estructura de eventos. Amazon S3 utiliza la versión 2.2 para las notificaciones de eventos de replicación entre regiones. Utiliza la versión 2.3 para S3 Lifecycle, S3 Intelligent-Tiering, la ACL de objetos, el etiquetado de objetos y los eventos de eliminación de restauración de objetos. Estas versiones contienen información adicional específica de estas operaciones. Las versiones 2.2 y 2.3 son compatibles con la versión 2.1, que Amazon S3 utiliza actualmente para todos los demás tipos de notificaciones de eventos.

```
{
   "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"
           }
        }
     }
```

```
    ]
}
```

Tenga en cuenta lo siguiente en relación con la estructura de mensajes de eventos:

- El valor de clave `eventVersion` contiene una versión principal y una versión secundaria con el formato `<major>.<minor>`.

  La versión principal se incrementa si Amazon S3 realiza un cambio en la estructura del evento que no es compatible con versiones anteriores. Esto incluye eliminar un campo JSON existente o cambiar la forma en que se representa el contenido de un campo (por ejemplo: un formato de fecha).

  La versión secundaria se incrementa si Amazon S3 añade campos a la estructura del evento. Esto puede ocurrir si se proporciona información nueva para algunos o todos los eventos existentes. Esto también puede ocurrir si se proporciona información nueva solo sobre los tipos de eventos recientemente introducidos. Las aplicaciones deben pasar por alto los campos nuevos para mantener la compatibilidad con versiones secundarias posteriores de la estructura de eventos.

  Si se introducen tipos de eventos nuevos, pero la estructura del evento no se modifica de ninguna otra forma, la versión del evento no cambia.

  Para asegurarse de que las aplicaciones pueden analizar la estructura de eventos correctamente, le recomendamos que haga una comparación de igualdad con el número de la versión principal. Para asegurarse de que los campos previstos por la aplicación están presentes, también recomendamos realizar una comparación mayor o igual que con la versión secundaria.
- El `eventName` hace referencia a la lista de tipos de notificaciones de eventos, pero no contiene el prefijo `s3:`.
- El valor de clave `responseElements` es útil si desea realizar el seguimiento de una solicitud con AWS Support. `x-amz-request-id` y `x-amz-id-2` ayudan a Amazon S3 a rastrear una solicitud individual. Estos valores son los mismos que los que devuelve Amazon S3 en respuesta a la solicitud que inicia los eventos. De este modo, se pueden utilizar para emparejar el evento con la solicitud.
- La clave `s3` brinda información acerca del bucket y el objeto involucrados en el evento. El valor del nombre de la clave de objeto está codificado como URL. Por ejemplo, "red flower.jpg" se convierte en "red+flower.jpg" (Amazon S3 devuelve "`application/x-www-form-urlencoded`" como tipo de contenido en la respuesta).

- La clave `sequencer` permite determinar la secuencia de los eventos. No se garantiza que las notificaciones de eventos lleguen en el mismo orden en que se produjeron los eventos. Sin embargo, las notificaciones de eventos que crean objetos (PUT) y borran objetos contienen un `sequencer`. Se puede utilizar para determinar el orden de los eventos de una clave de objeto determinada.

  Si compara las cadenas de `sequencer` de dos notificaciones de eventos en la misma clave de objeto, la notificación del evento con el mayor valor hexadecimal de `sequencer` es el evento que se produjo más tarde. Si utiliza notificaciones de eventos para mantener otra base de datos u otro índice de los objetos de Amazon S3, le recomendamos que compare y almacene los valores de `sequencer` a medida que procesa cada notificación de evento.

  Tenga en cuenta lo siguiente:

  - No se puede utilizar `sequencer` para determinar el orden de los eventos en diferentes claves de objeto.
  - El secuenciador puede ser de diferentes longitudes. Por lo tanto, para comparar estos valores, primero agregue ceros a la derecha del valor más corto y, a continuación, realice una comparación lexicográfica.
- La clave `glacierEventData` solo es visible de los eventos `s3:ObjectRestore:Completed`.
- La clave `restoreEventData` contiene atributos relacionados con la solicitud de restauración.
- La clave `replicationEventData` solo es visible para los eventos de replicación.
- La clave `intelligentTieringEventData` solo es visible para los eventos de S3 Intelligent-Tiering.
- La clave `lifecycleEventData` solo es visible para los eventos de transición de S3 Lifecycle.

Mensajes de ejemplo

A continuación se muestran ejemplos de mensajes de notificación de eventos de Amazon S3.

Mensaje de prueba de Amazon S3

Después de configurar una notificación de evento en un bucket, Amazon S3 envía el siguiente mensaje de prueba.

```
{
   "Service":"Amazon S3",
   "Event":"s3:TestEvent",
```

```
   "Time":"2014-10-13T15:57:02.089Z",
   "Bucket":"bucketname",
   "RequestId":"5582815E1AEA5ADF",
   "HostId":"8cLeGAmw098X5cv4Zkwcmo8vvZa3eH3eKxsPzbB9wrR+YstdA6Knx4Ip8EXAMPLE"
}
```

Mensaje de ejemplo cuando se crea un objeto al utilizar una solicitud PUT

El siguiente mensaje es un ejemplo de un mensaje que Amazon S3 envía para publicar un `s3:ObjectCreated:Put` evento.

```
{
   "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"
            }
        }
    }
  ]
}
```

Para obtener una definición de cada prefijo de identificación de IAM (por ejemplo, AIDA, AROA, AGPA), consulte Identificadores de IAM) en la Guía del usuario de IAM.

Uso de EventBridge

Amazon S3 puede enviar eventos a Amazon EventBridge cada vez que se producen ciertos eventos en el bucket. A diferencia de otros destinos, no es necesario seleccionar qué tipos de eventos desea entregar. Una vez habilitado EventBridge, todos los eventos siguientes se envían a EventBridge. Puede utilizar las reglas de EventBridge para dirigir eventos a destinos adicionales. A continuación, se enumeran los eventos que Amazon S3 envía a EventBridge.

Tipo de evento	Descripción
Objeto creado	Se creó un objeto. El campo motivo de la estructura de mensajes de eventos indica qué API de S3 se utilizó para crear el objeto: PutObject, POST Object, CopyObject o CompleteMultipartUpload.
Objeto eliminado (DeleteObject) Objeto eliminado (vencimiento del ciclo de vida)	Se eliminó un objeto. Cuando se elimina un objeto mediante una llamada a la API de S3, el campo de motivo se establece en DeleteObject. Cuando una regla de vencimiento de S3 Lifecycle elimina un objeto, el campo de motivo se establece en Lifecycle Expiration (Vencimiento del ciclo de vida). Para obtener más información, consulte Vencimiento de objetos. Cuando se elimina un objeto sin control de versiones o se elimina de forma permanente un objeto con control de

Tipo de evento	Descripción
	versiones, el campo de tipo de eliminación se establece en Permanently Deleted (Eliminado de forma permanente). Cuando se crea un marcador de eliminación para un objeto con control de versiones, el campo de tipo de eliminaci ón se establece en Delete Marker Created (Marcador de eliminación creado). Para obtener más información, consulte Eliminar versiones de objetos de un bucket con control de versiones habilitado.
Restauración de objetos iniciada	Se inició una restauración de objetos desde la clase de almacenamiento S3 Glacier o S3 Glacier Deep Archive o desde el nivel S3 Intelligent-Tiering Archive Access o Deep Archive Access. Para obtener más información, consulte Trabajar con objetos archivados.
Restauración de objetos completad a	Se completó una restauración de objetos.
Restauración de objetos vencida	La copia temporal de un objeto restaurado desde S3 Glacier o S3 Glacier Deep Archive venció y se eliminó.
Cambio de clase de almacenam iento de objetos	Se cambió un objeto a una clase de almacenamiento diferente. Para obtener más información, consulte Transició n de objetos con Amazon S3 Lifecycle.
Cambio de nivel de acceso a objetos	Se cambió un objeto al nivel S3 Intelligent-Tiering Archive Access o Deep Archive Access. Para obtener más información, consulte Amazon S3 Intelligent Tiering.
ACL de objeto actualizada	Se configuró la lista de control de acceso (ACL) de un objeto mediante PutObjectACL. No se genera un evento cuando una solicitud no produce cambios en la ACL de un objeto. Para obtener más información, consulte Información general de las Listas de control de acceso (ACL).

Tipo de evento	Descripción
Etiquetas de objeto agregadas	Se agregó un conjunto de etiquetas a un objeto mediante PutObjectTagging. Para obtener más información, consulte Categorización del almacenamiento mediante etiquetas.
Etiquetas de objeto eliminadas	Se eliminaron todas las etiquetas de un objeto mediante DeleteObjectTagging. Para obtener más información, consulte Categorización del almacenamiento mediante etiquetas.

> **Note**
>
> Para obtener más información sobre cómo se asignan los tipos de eventos de Amazon S3 a los tipos de eventos de EventBridge, consulte Asignación y solución de problemas de Amazon EventBridge.

Puede utilizar las notificaciones de eventos de Amazon S3 con EventBridge para escribir reglas que realizan acciones cuando se produce un evento en el bucket. Por ejemplo, puede hacer que le envíe una notificación. Para obtener más información, consulte ¿Qué es EventBridge? en la Guía del usuario de Amazon EventBridge.

Para obtener más información sobre las acciones y los tipos de datos con los que puede interactuar mediante la API de EventBridge. consulte la Referencia de la API de Amazon EventBridge en la Referencia de la API de Amazon EventBridge.

Para obtener información acerca de los precios, consulte Precios de Amazon EventBridge.

Temas

- Permisos de Amazon EventBridge
- Activación de Amazon EventBridge
- Estructura de mensaje de evento EventBridge
- Asignación y solución de problemas de Amazon EventBridge

Permisos de Amazon EventBridge

Amazon S3 no requiere permisos adicionales para entregar eventos a Amazon EventBridge.

Activación de Amazon EventBridge

Puede habilitar Amazon EventBridge mediante la consola de S3, AWS Command Line Interface (AWS CLI) o REST de API de Amazon S3.

Uso de la consola de S3

Para habilitar la entrega de eventos de EventBridge en la consola de S3.

1. Inicie sesión AWS Management Console Management Console y abra la consola de Amazon S3 en https://console.aws.amazon.com/s3/.
2. En la lista Buckets, seleccione el nombre del bucket para el que desea habilitar eventos.
3. Seleccione Properties (Propiedades).
4. Vaya a la sección Event Notifications (Notificaciones de eventos) y busque la subsección Amazon EventBridge. Elija Editar.
5. En Send notifications to Amazon EventBridge for all events in this bucket (Enviar notificaciones a Amazon EventBridge para todos los eventos de este bucket), elija On (Activado).

> **Note**
>
> Después de habilitar EventBridge, los cambios tardan alrededor de cinco minutos en aplicarse.

Uso de la AWS CLI

En el siguiente ejemplo, se crea una configuración de notificación de bucket para el bucket `amzn-s3-demo-bucket1` con Amazon EventBridge habilitado.

```
aws s3api put-bucket-notification-configuration --bucket amzn-s3-demo-bucket1 --
notification-configuration='{ "EventBridgeConfiguration": {} }'
```

Uso de la API de REST

Puede habilitar Amazon EventBridge en un bucket mediante programación llamando a la API de REST de Amazon S3. Para obtener más información, consulte PutBucketNotificationConfiguration en la Referencia de la API de Amazon Simple Storage Service.

En el siguiente ejemplo, se muestra el XML utilizado para crear una configuración de notificación de bucket con Amazon EventBridge habilitado.

```
<NotificationConfiguration xmlns="http://s3.amazonaws.com/doc/2006-03-01/">
  <EventBridgeConfiguration>
  </EventBridgeConfiguration>
</NotificationConfiguration>
```

Creación de reglas de EventBridge

Una vez habilitado, puede crear reglas de Amazon EventBridge para determinadas tareas. Por ejemplo, puede enviar notificaciones por correo electrónico cuando se crea un objeto. Para ver un tutorial completo, consulte Tutorial: Envío de una notificación cuando se crea un objeto de Amazon S3 en la Guía del usuario de Amazon EventBridge.

Estructura de mensaje de evento EventBridge

El mensaje de notificación que Amazon S3 envía para publicar un evento está en formato JSON. Cuando Amazon S3 envía un evento a Amazon EventBridge, aparecen los siguientes campos.

- version (versión): actualmente 0 (cero) para todos los eventos.
- id: un UUID de versión 4 generado para cada evento.
- detail-type (tipo de detalle): el tipo de evento que se envía. Para obtener una lista de los tipos de eventos, consulte Uso de EventBridge.
- source (origen): identifica el servicio que generó el evento.
- account (cuenta): el ID de 12 dígitos de la Cuenta de AWS del propietario del bucket.
- time (hora): la hora en que ocurrió el evento.
- region (región): identifica la Región de AWS del bucket.
- resources (recursos): una matriz JSON que contiene el nombre de recurso de Amazon (ARN) del bucket.

- detail (detalle): un objeto JSON que contiene información sobre el evento. Para obtener más información acerca de lo que se puede incluir en este campo, consulte Campo de detalle del mensaje de evento.

Ejemplos de estructura de mensaje de evento

A continuación, se muestran ejemplos de algunos de los mensajes de notificación de eventos de Amazon S3 que se pueden enviar a Amazon EventBridge.

Objeto creado

```
{
  "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:::amzn-s3-demo-bucket1"
  ],
  "detail": {
    "version": "0",
    "bucket": {
      "name": "amzn-s3-demo-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"
  }
}
```

Objeto eliminado (utilizando 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:::amzn-s3-demo-bucket1"
  ],
  "detail": {
    "version": "0",
    "bucket": {
      "name": "amzn-s3-demo-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"
  }
}
```


Objeto eliminado (utilizando vencimiento del ciclo de vida)


```
{
  "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:::amzn-s3-demo-bucket1"
  ],
  "detail": {
    "version": "0",
    "bucket": {
      "name": "amzn-s3-demo-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"
  }
}
```

Restauración de objetos completada

```
{
  "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:::amzn-s3-demo-bucket1"
  ],
  "detail": {
    "version": "0",
    "bucket": {
      "name": "amzn-s3-demo-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"
  }
}
```

Campo de detalle del mensaje de evento

El campo de detalles contiene un objeto JSON con información sobre el evento. Los siguientes campos pueden estar presentes en el campo de detalles.

- version (versión): actualmente 0 (cero) para todos los eventos.
- bucket: información sobre el bucket de Amazon S3 involucrado en el evento.
- object (objeto): información sobre el objeto de Amazon S3 involucrado en el evento.
- request-id (id de solicitud): ID de solicitud en la respuesta de S3.
- requester (solicitante): ID de la Cuenta de AWS o entidad principal de servicio de AWS del solicitante.
- source-ip-address (dirección IP de origen): dirección IP de origen de la solicitud de S3. Solo está presente para eventos desencadenados por una solicitud de S3.
- reason (motivo): para eventos de objeto creado, la API de S3 solía crear el objeto: PutObject, POST Object, CopyObject o CompleteMultipartUpload. Para eventos de objeto eliminado, esto se configura en DeleteObject cuando se elimina un objeto mediante una llamada a la API de S3, o vencimiento del ciclo de vida cuando se elimina un objeto mediante una regla de vencimiento de S3 Lifecycle. Para obtener más información, consulte Vencimiento de objetos.
- deletion-type: para eventos de objeto eliminado, cuando se elimina un objeto sin control de versiones o se elimina de forma permanente un objeto con control de versiones, esto se establece en Permanently Deleted (Eliminado de forma permanente). Cuando se crea un marcador de eliminación para un objeto con control de versiones, esto se establece en Delete Marker Created (Marcador de eliminación creado). Para obtener más información, consulte Eliminar versiones de objetos de un bucket con control de versiones habilitado.

> **Note**
>
> Algunos atributos del objeto (como `etag` y `size`) solo están presentes cuando se crea un marcador de eliminación.

- restore-expiry-time (restauración de tiempo de vencimiento): para eventos de restauración de objetos completada, la hora en que la copia temporal del objeto se eliminará de S3. Para obtener más información, consulte Trabajar con objetos archivados.
- source-storage-class (clase de almacenamiento de origen): para eventos de restauración de objetos iniciada y restauración de objetos completada, la clase de almacenamiento del objeto que se está restaurando. Para obtener más información, consulte Trabajar con objetos archivados.
- destination-storage-class (clase de almacenamiento de destino): para eventos de cambio de clase de almacenamiento de objetos, la nueva clase de almacenamiento del objeto. Para obtener más información, consulte Transición de objetos con Amazon S3 Lifecycle.
- destination-access-tier (nivel de acceso de destino): para eventos de cambio de nivel de acceso a objetos, el nuevo nivel de acceso del objeto. Para obtener más información, consulte Amazon S3 Intelligent Tiering.

Asignación y solución de problemas de Amazon EventBridge

En la tabla siguiente, se describe cómo se asignan los tipos de eventos de Amazon S3 a los tipos de eventos de Amazon EventBridge.

Tipo de evento de S3	Tipo de detalle de Amazon EventBridge
ObjectCreated:Put ObjectCreated:Post ObjectCreated:Copy ObjectCreated:CompleteMultipartUpload	Objeto creado
ObjectRemoved:Delete	Objetos eliminados

Tipo de evento de S3	Tipo de detalle de Amazon EventBridge
ObjectRemoved:DeleteMarkerCreated LifecycleExpiration:Delete LifecycleExpiration:DeleteMarkerCreated	
ObjectRestore:Post	Restauración de objetos iniciada
ObjectRestore:Completed	Restauración de objetos completada
ObjectRestore:Delete	Restauración de objetos vencida
LifecycleTransition	Cambio de clase de almacenamiento de objetos
IntelligentTiering	Cambio de nivel de acceso a objetos
ObjectTagging:Put	Etiquetas de objeto agregadas
ObjectTagging:Delete	Etiquetas de objeto eliminadas
ObjectAcl:Put	ACL de objeto actualizada

Solución de problemas de Amazon EventBridge

Para obtener información acerca de cómo solucionar problemas de EventBridge, consulte Solución de problemas de Amazon EventBridge en la Guía del usuario de Amazon EventBridge.

Uso de análisis e información

Puede utilizar el análisis y la información en Amazon S3 a fin de comprender, analizar y optimizar el uso del almacenamiento. Para obtener más información, consulte los siguientes temas.

Temas

Análisis de Amazon S3: análisis de clases de almacenamiento

Con la herramienta de análisis de clases de almacenamiento de Amazon S3 podrá analizar los patrones de acceso al almacenamiento para poder determinar cuándo trasladar los datos adecuados según la clase de almacenamiento apropiada. Esta nueva función de análisis de las clases de almacenamiento de Amazon S3 observa los patrones de acceso a los datos para ayudarle a determinar cuándo trasladar el almacenamiento STANDARD al que se acceda con menos frecuencia a la clase de almacenamiento STANDARD_IA (IA hace referencia a “acceso poco frecuente”). Para obtener más información acerca de las clases de almacenamiento, consulte Uso de las clases de almacenamiento de Amazon S3.

Después de que el análisis de clase de almacenamiento observe estos patrones de acceso poco frecuentes de un conjunto de datos filtrados durante un periodo determinado de tiempo, puede usar los resultados del análisis para mejorar la configuración del ciclo de vida. Puede configurar el análisis de clases de almacenamiento para analizar todos los objetos de un bucket. También puede configurar filtros para agrupar objetos de modo que se analicen según un prefijo común (es decir, objetos que tengan nombres que comiencen por una cadena común), por etiquetas de objetos o por prefijos y etiquetas. Seguramente descubrirá que el filtrado por grupos de objetos es la mejor manera de beneficiarse del análisis de clases de almacenamiento.

> **Important**
>
> El análisis de clases de almacenamiento solo proporciona recomendaciones para las clases Standard a Standard-IA.

Puede tener varios análisis de clases de almacenamiento por bucket, hasta 1000, y recibirá un análisis separado para cada filtro. Las configuraciones de múltiples filtros le permiten analizar grupos específicos de objetos para mejorar las configuraciones de ciclo de vida que pasen objetos a STANDARD IA.

El análisis de clases de almacenamiento muestra visualizaciones del uso en la consola de Amazon S3 que se actualizan a diario. También puede exportar estos datos de uso diario a un bucket de S3 y verlos en una aplicación de hoja de cálculo, o con herramientas de inteligencia empresarial, como Amazon QuickSight.

Hay costos asociados con el análisis de clases de almacenamiento. Para obtener información sobre precios, consulte Administración y replicación en Precios de Amazon S3.

Temas

¿Cómo se configura el análisis de clases de almacenamiento?

Puede configurar el análisis de clases de almacenamiento configurando qué datos de objetos quiere analizar. Puede configurar el análisis de clases de almacenamiento para hacer lo siguiente:

- Analizar la totalidad de los contenidos de un bucket.

  Recibirá el análisis de todos los objetos del bucket.
- Analizar objetos agrupados por prefijos y etiquetas.

  Puede configurar filtros para agrupar objetos de modo que se analicen según un prefijo común, por etiquetas de objetos o por prefijos y etiquetas. Recibirá un análisis separado para cada filtro que configure. Puede tener varias configuraciones de filtro por cada bucket, hasta 1000.
- Exportar análisis de datos

  Al configurar el análisis de clases de almacenamiento para un bucket o un filtro, puede decidir que los datos del análisis se exporten a un archivo a diario. El análisis diario se agrega al archivo para formar un registro de análisis histórico para el filtro configurado. El archivo se actualiza a diario en

el destino de su elección. Al seleccionar qué datos exportar, especifique un bucket de destino y un prefijo de destino opcional donde se escribirá el archivo.

Puede utilizar la consola de Amazon S3, la API REST, la AWS CLI o los SDK de AWS para configurar el análisis de clases de almacenamiento.

- Para obtener información acerca de cómo configurar el análisis de clase de almacenamiento en la consola de Amazon S3, consulte Configuración del análisis de clases de almacenamiento.
- Para utilizar la API de Amazon S3, use la API REST PutBucketAnalyticsConfiguration o su equivalente desde la AWS CLI o los SDK de AWS.

¿Cómo se usa el análisis de clases de almacenamiento?

Puede usar el análisis de clases de almacenamiento para observar los patrones de acceso a los datos a lo largo del tiempo para reunir información que le ayude a mejorar la administración del ciclo de vida de su almacenamiento STANDARD_IA. Tras configurar un filtro, empezará a ver un análisis de datos en función del filtro en la consola de Amazon S3 en un plazo de 24 a 48 horas. Sin embargo, el análisis de clases de almacenamiento observa los patrones de acceso de un conjunto de datos filtrado durante 30 o más días para reunir información y analizarla antes de dar un resultado. El análisis sigue ejecutándose tras el resultado inicial y actualiza el resultado a medida que cambian los patrones de acceso.

Cuando configura un filtro por primera vez, la consola de Amazon S3) puede tardar un momento en analizar los datos.

El análisis de clases de almacenamiento observa los patrones de acceso de un conjunto de datos de objetos filtrado durante 30 o más días para reunir información suficiente para el análisis. Después de que el análisis de clases de almacenamiento haya reunido la información suficiente, verá un mensaje en la consola de Amazon S3 que indica que se completó el análisis.

Al realizar un análisis de clase de almacenamiento para objetos a los que se accede con poca frecuencia, el análisis se dirige a un conjunto filtrado de objetos agrupados según su edad desde que se cargaron en Amazon S3. El análisis de clases de almacenamiento determina si se obtiene acceso al grupo de edad con poca frecuencia analizando los siguientes factores para el conjunto de datos filtrado:

- Los objetos en la clase de almacenamiento STANDARD son mayores de 128 KB.
- Cuánto almacenamiento total medio tiene por cada grupo de edad.

- Número medio de bytes transferidos (sin frecuencia) por cada grupo de edad.
- Los datos de análisis exportados solo incluyen solicitudes que contengan datos relevantes para el análisis de clases de almacenamiento. Esto podría provocar que surjan diferencias en el número de solicitudes, y en el total de bytes de carga y solicitudes comparado con lo que se muestra en las métricas de almacenamiento o con el seguimiento de sus propios sistemas internos.
- Las solicitudes GET y PUT que den error no se cuentan para el análisis. Sin embargo, sí que se ven estas solicitudes erróneas en las métricas de almacenamiento.

¿Qué cantidad de almacenamiento he recuperado?

La consola de Amazon S3 ilustra la cantidad de almacenamiento del conjunto de datos filtrado que se ha recuperado para el periodo de observación.

¿Qué porcentaje de almacenamiento he recuperado?

La consola de Amazon S3 también ilustra el porcentaje de almacenamiento del conjunto de datos filtrado que se ha recuperado para el periodo de observación.

Como ya hemos afirmado antes en este tema, al realizar un análisis de clase de almacenamiento para objetos a los que se accede con poca frecuencia, el análisis se dirige a un conjunto filtrado de objetos agrupados según su edad desde que se cargaron en Amazon S3. El análisis de clases de almacenamiento emplea los siguientes grupos de edad de objetos predefinidos:

- Objetos de Amazon S3 de menos de 15 días
- Objetos de Amazon S3 de 15 a 29 días de antigüedad
- Objetos de Amazon S3 de 30 a 44 días de antigüedad
- Objetos de Amazon S3 de 45 a 59 días de antigüedad
- Objetos de Amazon S3 de 60 a 74 días de antigüedad
- Objetos de Amazon S3 de 75 a 89 días de antigüedad
- Objetos de Amazon S3 de 90 a 119 días de antigüedad
- Objetos de Amazon S3 de 120 a 149 días de antigüedad
- Objetos de Amazon S3 de 150 a 179 días de antigüedad
- Objetos de Amazon S3 de 180 a 364 días de antigüedad
- Objetos de Amazon S3 de 365 a 729 días de antigüedad
- Objetos de Amazon S3 de 730 días o más

Normalmente, se necesita observar los patrones de acceso durante unos 30 días para reunir la información suficiente para el resultado de un análisis. Podría llevar más de 30 días, en función del patrón de acceso exclusivo de sus datos. No obstante, tras configurar un filtro, empezará a ver un análisis de datos según el filtro en la consola de Amazon S3 en un plazo de 24 a 48 horas. Puede ver el análisis de acceso a los objetos por días desglosados por grupos de edad del objeto en la consola de Amazon S3.

¿A qué cantidad de mi almacenamiento se obtiene acceso con poca frecuencia?

La consola de Amazon S3 muestra los patrones de acceso agrupados por los grupos de edad de objetos predefinidos. El texto se acceso frecuente o de acceso infrecuente que se muestra es una ayuda visual para ayudarle en el proceso de creación del ciclo de vida.

¿Cómo se pueden exportar los datos del análisis de clases de almacenamiento?

Puede decidir que el análisis de clases de almacenamiento exporte los informes de análisis en un archivo plano con formato de valores separados por comas (CSV). Los informes se actualizan a diario y se basan en los filtros de grupos de edad de los objetos que configure. Al usar la consola de Amazon S3 puede seleccionar la opción de exportar informes al crear un filtro. Al seleccionar qué datos exportar, especifique un bucket de destino y un prefijo de destino opcional donde se escribirá el archivo. Puede exportar los datos en un bucket de destino en una cuenta diferente. El bucket de destino debe estar en la misma región que el bucket que configure para el análisis.

Debe crear una política de bucket en el bucket de destino para conceder permisos a fin de que Amazon S3 compruebe qué Cuenta de AWS es propietaria del bucket y escriba objetos en el bucket en la ubicación definida. Para ver una política de ejemplo, consulte Concesión de permisos para el inventario de S3 y el análisis de S3.

Tras configurar los informes de análisis de las clases de almacenamiento, empezará a recibir el informe exportado a diario tras 24 horas. Después de ese momento, Amazon S3 seguirá monitoreando y proporcionando exportaciones diarias.

Puede abrir el archivo CSV en una aplicación de hoja de cálculo o importar el archivo en otras aplicaciones, como Amazon Quicksight. Para obtener información sobre cómo usar archivos de Amazon S3 con Amazon QuickSight, consulte Creación de un conjunto de datos utilizando archivos deAmazon S3 en la guía de usuario de Amazon QuickSight.

Los datos en el archivo exportado se ordenan por fecha en los grupos de edad de objetos, como se muestra en los siguientes ejemplos. Si la clase de almacenamiento es STANDARD,

la fila contendrá también datos para las columnas `ObjectAgeForSIATransition` y `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		

Al final del informe, el grupo de edad de los objetos se proporciona como ALL. Las filas ALL contienen totales acumulados, incluidos los objetos inferiores a 128 KB, para todos los grupos de edad de ese día.

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

En la siguiente sección se describen las columnas que se usan en el informe.

Diseñar el archivo exportado

En la siguiente tabla se describe el diseño del archivo exportado.

Configuración del análisis de clases de almacenamiento

Al usar la herramienta de análisis de clases de almacenamiento de Amazon S3, podrá analizar los patrones de acceso al almacenamiento para poder determinar cuándo trasladar los datos apropiados a la clase de almacenamiento apropiada. El análisis de las clases de almacenamiento observa los patrones de acceso a los datos para ayudarle a determinar cuándo trasladar el almacenamiento STANDARD al que se acceda con menos frecuencia a la clase de almacenamiento STANDARD_IA (IA quiere decir acceso poco frecuente). Para obtener más información acerca de STANDARD_IA, consulte las preguntas frecuentes de Amazon S3 y Uso de las clases de almacenamiento de Amazon S3.

Puede configurar el análisis de clases de almacenamiento configurando qué datos de objetos quiere analizar. Puede configurar el análisis de clases de almacenamiento para hacer lo siguiente:

- Analizar la totalidad de los contenidos de un bucket.

  Recibirá el análisis de todos los objetos del bucket.
- Analizar objetos agrupados por prefijos y etiquetas.

Puede configurar filtros para agrupar objetos de modo que se analicen según un prefijo común, por etiquetas de objetos o por prefijos y etiquetas. Recibirá un análisis separado para cada filtro que configure. Puede tener varias configuraciones de filtro por cada bucket, hasta 1000.

- Exportar análisis de datos

  Al configurar el análisis de clases de almacenamiento para un bucket o un filtro, puede decidir que los datos del análisis se exporten a un archivo a diario. El análisis diario se agrega al archivo para formar un registro de análisis histórico para el filtro configurado. El archivo se actualiza a diario en el destino de su elección. Al seleccionar qué datos exportar, especifique un bucket de destino y un prefijo de destino opcional donde se escribirá el archivo.

Puede utilizar la consola de Amazon S3, la API REST, la AWS CLI o los SDK de AWS para configurar el análisis de clases de almacenamiento.

> **Important**
>
> El análisis de clases de almacenamiento no ofrece recomendaciones para las transiciones a las clases de almacenamiento ONEZONE_IA o S3 Glacier Flexible Retrieval.
> Si desea configurar el análisis de clase de almacenamiento para exportar sus conclusiones como un archivo .csv y el bucket de destino utiliza el cifrado de bucket predeterminado con una AWS KMS key, debe actualizar la política de claves de AWS KMS a fin de conceder permiso a Amazon S3 para cifrar el archivo .csv. Para obtener instrucciones, consulte Concesión de permiso a Amazon S3 con el fin de utilizar su clave administrada por el cliente para el cifrado.

Para obtener más información sobre los análisis, consulte Análisis de Amazon S3: análisis de clases de almacenamiento.

Uso de la consola de S3

Para configurar el análisis de clases de almacenamiento

1. Inicie sesión en la AWS Management Console y abra la consola de Amazon S3 en https://console.aws.amazon.com/s3/.
2. En la lista Buckets, seleccione el nombre del bucket para el que desea configurar el análisis de clases de almacenamiento.

3. Elija la pestaña Metrics (Métricas).
4. En Storage Class Analysis (Análisis de clases de almacenamiento), elija Create analytics configuration (Crear configuración de análisis).
5. Escriba un nombre para el filtro. Si quiere analizar todo el bucket, deje el campo Prefix (Prefijo) vacío.
6. En el campo Prefix (Prefijo), escriba texto para el prefijo de los objetos que desee analizar.
7. Para agregar una etiqueta, elija Add tag (Añadir etiqueta). Escriba una clave y un valor para la etiqueta. Puede introducir un prefijo y varias etiquetas.
8. De manera opcional, también puede seleccionar Enable (Habilitar) en Export CSV (Exportar CSV) para exportar los informes de análisis en un archivo plano de valores separados por comas (.csv). Seleccione un bucket de destino donde poder guardar el archivo. Puede escribir un prefijo para el bucket de destino. El bucket de destino debe estar en la misma Región de AWS que el bucket para el que configura el análisis. El bucket de destino puede estar en una Cuenta de AWS diferente.

   Si el bucket de destino del archivo .csv utiliza el cifrado de bucket predeterminado con una clave de KMS, debe actualizar la política de claves de AWS KMS a fin de conceder permiso a Amazon S3 para cifrar el archivo .csv. Para obtener instrucciones, consulte Concesión de permiso a Amazon S3 con el fin de utilizar su clave administrada por el cliente para el cifrado.
9. Seleccione Create configuration (Crear configuración).

Amazon S3 crea una política de bucket en el bucket de destino que concede permisos de escritura a Amazon S3. Esto le permite escribir los datos de exportación en el bucket.

Si se produce un error al intentar crear la política de bucket, recibirá instrucciones para solucionarlo. Por ejemplo, si elige un bucket de destino en otra Cuenta de AWS y no tiene permisos para leer y escribir en la política del bucket, verá el siguiente mensaje. Debe hacer que el propietario del bucket de destino agregue la política de bucket que se muestra en el bucket de destino. Si la política no se agrega en el bucket de destino, no obtendrá los datos de exportación, ya que Amazon S3 no tiene permiso para escribir en el bucket de destino. Si el bucket de origen es propiedad de una cuenta diferente de la del usuario actual, el ID de cuenta correcto del bucket de origen debe sustituirse en la política.

Para obtener información sobre los datos exportados y cómo funciona el filtro, consulte Análisis de Amazon S3: análisis de clases de almacenamiento.

Uso de la API REST

Para configurar el análisis de clases de almacenamiento mediante la API REST, utilice PutBucketAnalyticsConfiguration. También puede utilizar la operación equivalente con la AWS CLI o los SDK de AWS.

Puede utilizar las siguientes API REST para trabajar con el análisis de clases de almacenamiento:

- DELETE Bucket Analytics configuration
- GET Bucket Analytics configuration
- List Bucket Analytics Configuration

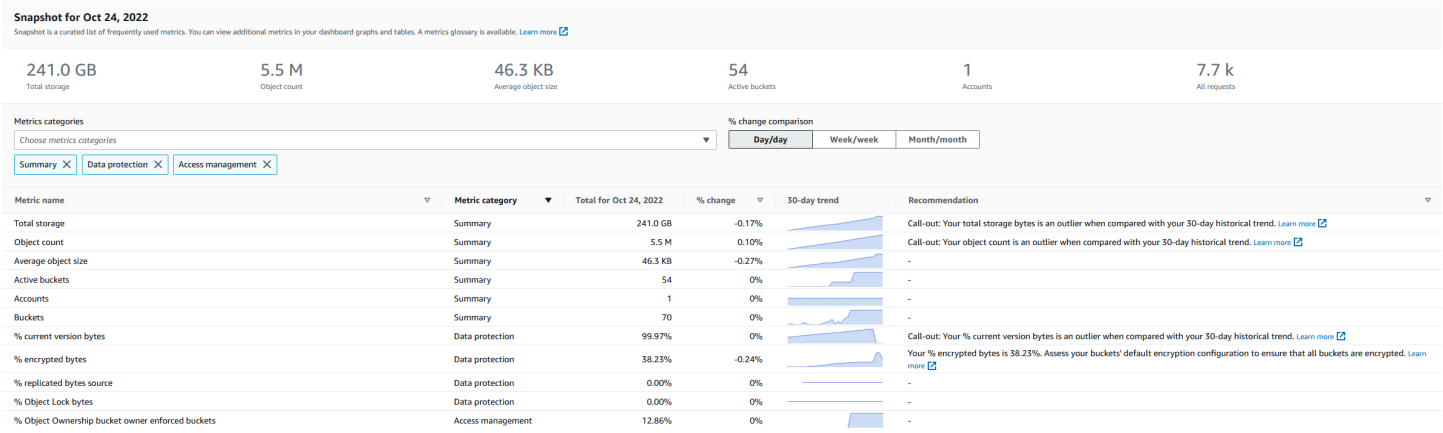
Evaluación de la actividad y el uso de almacenamiento con Amazon S3 Storage Lens

Amazon S3 Storage Lens es una característica de análisis de almacenamiento en la nube que puede utilizar para obtener visibilidad de toda la organización sobre el uso y la actividad del almacenamiento de objetos. S3 Storage Lens analiza también las métricas para ofrecer recomendaciones contextuales que puede usar para optimizar los costos de almacenamiento y aplicar las prácticas recomendadas para proteger los datos.

Puede usar métricas de Lente de almacenamiento de S3 para generar información resumida. Por ejemplo, puede averiguar cuánto almacenamiento tiene en toda la organización o cuáles son los buckets y los prefijos de crecimiento más rápido. También puede utilizar las métricas de Lente de almacenamiento de S3 para identificar oportunidades de optimización de costos, implementar las prácticas recomendadas de protección de datos y administración de acceso y mejorar el rendimiento de las cargas de trabajo de las aplicaciones. Por ejemplo, puede identificar los buckets que no tienen reglas del ciclo de vida de S3 para que aborten las cargas multipartes incompletas que tengan más de 7 días de antigüedad. También puede identificar los buckets que no siguen las prácticas recomendadas de protección de datos, como el uso de la Replicación de S3 o el control de versiones de S3.

Lente de almacenamiento de S3 agrega las métricas y muestra la información en la sección Account snapshot (Instantánea de la cuenta) en la página Buckets de la consola de Amazon S3. S3 Storage Lens también proporciona un panel interactivo que puede utilizar para visualizar información y tendencias, marcar valores atípicos y recibir recomendaciones para optimizar los costes de almacenamiento y aplicar las prácticas recomendadas de protección de datos. El panel tiene

opciones de desglose para generar y visualizar información en el nivel de la organización, la cuenta, la Región de AWS, la clase de almacenamiento, el bucket, el prefijo o el grupo de Lente de almacenamiento. También puede enviar una exportación de métricas diaria en formato CSV o Parquet a un bucket de S3.



Métricas y características de Lente de almacenamiento de S3

Lente de almacenamiento de S3 proporciona un panel predeterminado interactivo que se actualiza diariamente. Lente de almacenamiento de S3 preconfigura este panel para visualizar la información resumida y las tendencias de toda la cuenta y las actualiza diariamente en la consola de S3. Las métricas para este panel también se resumen en la instantánea de la cuenta en la página de Buckets. Para obtener más información, consulte Panel predeterminado.

Para crear otros paneles y segmentarlos por Regiones de AWS, buckets de S3 o cuentas (para AWS Organizations), se crea una configuración de panel de Lente de almacenamiento de S3. Puede crear y administrar las configuraciones de panel de S3 Storage Lens mediante AWS Command Line Interface, (AWS CLI), los SDK de AWS o la API de REST. Cuando crea o edita un panel de Lente de almacenamiento de S3, define el alcance del panel y la selección de métricas.

Lente de almacenamiento de S3 ofrece métricas gratuitas y avanzadas y recomendaciones, que puede actualizar por un cargo adicional. Con métricas y recomendaciones avanzadas, puede acceder a métricas y características adicionales para obtener información sobre el almacenamiento. Estas características incluyen las categorías de métricas avanzadas, la agregación de prefijos, las recomendaciones contextuales y la publicación en Amazon CloudWatch. La agregación de prefijos y las recomendaciones contextuales solo están disponibles en la consola de Amazon S3. Para obtener información acerca de los precios de S3 Storage Lens, consulte Precios de Amazon S3.

Categorías de métricas

Dentro de los niveles gratuito y avanzado, las métricas se organizan en categorías que se alinean con los casos de uso clave, como la optimización de costos y la protección de datos. Las métricas gratuitas incluyen métricas de resumen, optimización de costos, protección de datos, gestión de acceso, rendimiento y eventos. Al actualizar a métricas y recomendaciones avanzadas, puede habilitar métricas de optimización de costos y protección de datos avanzadas. Puede utilizar estas métricas avanzadas para reducir aún más los costos de almacenamiento de S3 y mejorar la postura de protección de datos. También puede habilitar las métricas de actividad y las métricas de código de estado detalladas para mejorar el rendimiento de las cargas de trabajo de las aplicaciones que acceden a los buckets de S3. Para obtener más información sobre las categorías de métricas gratuitas y avanzadas, consulte Selección de métricas.

Puede evaluar el almacenamiento en función de las prácticas recomendadas de S3, como analizar el porcentaje de buckets que tienen habilitado el cifrado, S3 Object Lock o el control de versiones de S3. También puede identificar oportunidades de ahorro de costos potenciales. Por ejemplo, puede utilizar las métricas del recuento de reglas del ciclo de vida de S3 para identificar los buckets a los que les faltan reglas de vencimiento del ciclo de vida o de transición. Puede también analizar la actividad de solicitud por bucket para encontrar buckets en los que los objetos podrían pasar a una clase de almacenamiento de menor costo. Para obtener más información, consulte Casos de uso de métricas de lente de almacenamiento de Amazon S3.

Exportación de métricas

Además de consultar el panel en la consola de S3, puede exportar métricas en formato CSV o Parquet a un bucket de S3 para su análisis posterior con la herramienta de análisis de su elección. Para obtener más información, consulte Visualización de las métricas de Amazon S3 Storage Lens mediante una exportación de datos.

Publicación de Amazon CloudWatch

Puede publicar métricas de actividad y uso de S3 Storage Lens en Amazon CloudWatch para crear una vista unificada del estado operativo en los paneles de CloudWatch. También puede utilizar las características de CloudWatch, como alarmas y acciones desencadenadas, cálculos de métricas y detección de anomalías para monitorear y tomar medidas en las métricas de Lente de almacenamiento de S3. Además, las operaciones de la API de CloudWatch habilitan las aplicaciones, incluidos los proveedores de terceros, para acceder a las métricas de Lente de almacenamiento de S3. La opción de publicación de CloudWatch está disponible para los paneles que se actualizan a métricas y recomendaciones avanzadas de Lente de almacenamiento de S3. Para obtener más información acerca del soporte para las métricas de S3 Storage Lens en CloudWatch, consulte Monitoreo de métricas de S3 Storage Lens en CloudWatch.

Para obtener más información acerca del uso de Lente de almacenamiento de S3, consulte los temas siguientes.

Temas

Compresión de Amazon S3 Storage Lens

> **Important**
>
> Amazon S3 aplica ahora el cifrado del servidor con claves administradas por Amazon S3 (SSE-S3) como el nivel básico de cifrado para cada bucket de Amazon S3. Desde el 5 de enero de 2023, todas las cargas de objetos nuevos a Amazon S3 se cifran automáticamente sin costo adicional y sin afectar al rendimiento. El estado de cifrado automático para la configuración de cifrado predeterminada en el bucket de S3 y para cargas de objetos nuevos está disponible en registros de AWS CloudTrail, Inventario de S3, Lente de almacenamiento de S3, la consola de Amazon S3 y como encabezado de respuesta a la API de Amazon S3 adicional en AWS Command Line Interface y los SDK de AWS. Para obtener más información, consulte Preguntas frecuentes del cifrado predeterminado.

Amazon S3 Storage Lens es una función de análisis de almacenamiento en la nube que puede utilizar para obtener visibilidad en toda la organización sobre el uso y la actividad del almacenamiento de objetos. Puede utilizar las métricas de S3 Storage Lens para generar información resumida, como averiguar cuánto almacenamiento tiene en toda la organización o cuáles son los buckets y los prefijos de crecimiento más rápido. También puede utilizar las métricas de Amazon S3 Storage Lens para identificar oportunidades de optimización de costos, implementar las prácticas recomendadas de protección y seguridad de los datos y mejorar el rendimiento de las cargas de

trabajo de las aplicaciones. Por ejemplo, puede identificar los buckets que no tienen reglas del ciclo de vida de S3 para que hagan vencer las cargas multipartes incompletas que tengan más de 7 días de antigüedad. También puede identificar los buckets que no siguen las prácticas recomendadas de protección de datos, como el uso de la Replicación de S3 o el control de versiones de S3. S3 Storage Lens analiza también las métricas para ofrecer recomendaciones contextuales que puede usar para optimizar los costos de almacenamiento y aplicar las prácticas recomendadas para proteger los datos.

S3 Storage Lens agrega las métricas y muestra la información en la sección Instantánea de la cuenta en la página Buckets de la consola de Amazon S3. S3 Storage Lens también proporciona un panel interactivo que puede utilizar para visualizar información y tendencias, marcar valores atípicos y recibir recomendaciones para optimizar los costes de almacenamiento y aplicar las prácticas recomendadas de protección de datos. El panel tiene opciones de desglose para generar y visualizar información en el nivel de la organización, la cuenta, la Región de AWS, la clase de almacenamiento, el bucket, el prefijo o el grupo de Lente de almacenamiento. También puede enviar una exportación de métricas diaria en formato CSV o Parquet a un bucket de S3. Puede crear y administrar los paneles de S3 Storage Lens mediante la consola de Amazon S3, AWS Command Line Interface, (AWS CLI), los SDK de AWS o la API de REST de Amazon S3.

Conceptos y terminología sobre Lente de almacenamiento de S3

Esta sección contiene la terminología y los conceptos esenciales para comprender y utilizar correctamente la Lente de almacenamiento de Amazon S3.

Temas

- S3 Storage Lens y AWS Organizations

Configuración del panel

Lente de almacenamiento de S3 requiere una configuración de panel que contenga las propiedades que se utilizan para agregar métricas en su nombre para un solo panel o exportación. Al crear una configuración, se elige el nombre del panel y la región de origen, que no puede cambiar después de crear el panel. Si lo desea, puede agregar etiquetas y configurar una exportación de métricas en formato CSV o Parquet.

En la configuración del panel, también se definen el alcance del panel y la selección de métricas. El alcance puede incluir todo el almacenamiento de la cuenta o las secciones de la organización filtradas por región, bucket y cuenta. Cuando se configura la selección de métricas, se elige entre métricas gratuitas y métricas y recomendaciones avanzadas, que puede actualizar con un costo adicional. Con métricas y recomendaciones avanzadas, puede acceder a métricas y características adicionales. Estas características incluyen las categorías de métricas avanzadas, la agregación de nivel de prefijo, las recomendaciones contextuales y la publicación de Amazon CloudWatch. Para obtener información acerca de los precios de S3 Storage Lens, consulte Precios de Amazon S3.

Panel predeterminado

El panel predeterminado de S3 Storage Lens en la consola se denomina default-account-dashboard. S3 preconfigura este panel para visualizar la información resumida y las tendencias de toda la cuenta y las actualiza diariamente en la consola de S3. No puede modificar el alcance de la configuración del panel predeterminado, pero puede actualizar la selección de métricas de métricas gratuitas a métricas y recomendaciones avanzadas. Puede configurar la exportación de métricas opcionales o incluso desactivar el panel. Sin embargo, no puede eliminar el panel predeterminado.

> **Note**
>
> Si desactiva el panel predeterminado, ya no se actualizará. Ya no recibirá ninguna métrica diaria nueva en el panel de S3 Storage Lens, la exportación de las métricas ni en la instantánea de la cuenta en la página Buckets de S3. Si en el panel se utilizan métricas y recomendaciones avanzadas, ya no se le cobrará. Aún podrá ver los datos históricos en el panel hasta el periodo de 14 días para que venzan las consultas de datos. Este periodo es de 15 meses si ha habilitado las métricas y recomendaciones avanzadas. Para acceder a datos históricos, puede volver a habilitar el panel dentro del periodo de vencimiento.

Paneles

Puede crear paneles de Lente de almacenamiento de S3 adicionales y segmentarlos por Regiones de AWS, buckets de S3 o cuentas (para AWS Organizations). Cuando crea o edita un panel de Lente de almacenamiento de S3, define el alcance del panel y la selección de métricas. Lente de almacenamiento de S3 ofrece métricas gratuitas y avanzadas y recomendaciones, que puede actualizar por un cargo adicional. Con métricas y recomendaciones avanzadas, puede acceder a métricas y características adicionales para obtener información sobre el almacenamiento. Entre ellas se incluyen las categorías de métricas avanzadas, la agregación en el nivel de prefijos, las recomendaciones contextuales y la publicación de Amazon CloudWatch. Para obtener información acerca de los precios de S3 Storage Lens, consulte Precios de Amazon S3.

También puede desactivar o eliminar paneles. Si deshabilita el panel, ya no se actualizará y ya no recibirá métricas diarias nuevas. Aún podrá ver los datos históricos hasta el periodo de vencimiento de 14 días. Si ha habilitado las métricas y recomendaciones avanzadas para ese panel, este periodo es de 15 meses. Para acceder a datos históricos, puede volver a habilitar el panel dentro del periodo de vencimiento.

Si elimina el panel, perderá todas las opciones de configuración del panel. Ya no recibirá métricas diarias nuevas y también perderá el acceso a los datos históricos asociados a ese panel. Si desea acceder a los datos históricos de un panel eliminado, debe crear otro panel con el mismo nombre en la misma región principal.

> **Note**
>
> - Puede utilizar S3 Storage Lens para crear hasta 50 paneles por cada región de origen.
> - Los paneles de nivel de organización se pueden limitar solo a un alcance regional.

Instantánea de cuenta

La Account snapshot (Instantánea de la cuenta) de Lente de almacenamiento de S3 resume las métricas del panel predeterminado y muestra el almacenamiento total, el recuento de objetos y el tamaño promedio de los objetos en la página de la consola de S3 Buckets. Esta instantánea de cuenta le brinda acceso rápido a información sobre el almacenamiento sin tener que salir de la página Buckets. La instantánea de la cuenta también permite acceder con un solo clic al panel interactivo de Lente de almacenamiento de S3.

Puede utilizar el panel para visualizar la información y las tendencias, marcar los valores atípicos y obtener las recomendaciones necesarias para optimizar los costos de almacenamiento y aplicar las prácticas recomendadas sobre protección de datos. El panel tiene opciones de desglose para generar información en el nivel de la organización, la cuenta, el bucket, el objeto o el prefijo. También puede enviar una exportación de métricas una vez al día a un bucket de S3 en formato CSV o Parquet.

No puede modificar el alcance del default-account dashboard (Panel de la cuenta predeterminado) porque está vinculado a la Account snapshot (Instantánea de la cuenta). Sin embargo, puede actualizar la selección de métricas en el default-account-dashboard de las métricas gratuitas a las métricas y recomendaciones avanzadas pagadas. Después de la actualización, puede mostrar todas las solicitudes, los bytes cargados y los bytes descargados en la Account snapshot (Instantánea de cuenta) de Lente de almacenamiento de S3.

> **Note**
>
> Si desactiva el panel predeterminado, la Account snapshot (Instantánea de la cuenta) ya no se actualizará. Para seguir mostrando métricas en la Account snapshot (Instantánea de la cuenta), puede volver a habilitar default-account-dashboard.

Exportación de métricas

Una exportación de métricas de S3 Storage Lens es un archivo que contiene todas las métricas identificadas en la configuración de S3 Storage Lens. Esta información se genera diariamente en formato CSV o Parquet y se envía a un bucket de S3. Puede utilizar la exportación de métricas para realizar un análisis más detallado mediante la herramienta de métricas que elija. El bucket de S3 para la exportación de métricas debe estar en la misma región que la configuración de la lente de almacenamiento S3. Puede generar una exportación de métricas de Lente de almacenamiento de S3 desde la consola de S3 si edita la configuración del panel. También puede configurar una exportación de métricas mediante la AWS CLI y los SDK de AWS.

Región de origen

La región principal es la Región de AWS donde todas las métricas de Lente de almacenamiento de S3 para una configuración de panel determinada se almacenan. Debe elegir una región de origen cuando cree la configuración del panel de Lente de almacenamiento de S3. Una vez que haya elegido una región de origen, no podrá cambiarla. Además, si va a crear un grupo de Lente de

almacenamiento, le recomendamos que elija la misma región de origen que su panel de Lente de almacenamiento.

> **Note**
>
> Puede elegir una de las siguientes regiones como región de origen:
>
> - Este de EE. UU. (Norte de Virginia) – `us-east-1`
> - Este de EE. UU. (Ohio) – `us-east-2`
> - Oeste de EE. UU. (Norte de California) – `us-west-1`
> - Oeste de EE. UU. (Oregón) – `us-west-2`
> - Asia-Pacífico (Bombay) – `ap-south-1`
> - Asia-Pacífico (Seúl) (`ap-northeast-2`)
> - Asia Pacífico (Singapur) – `ap-southeast-1`
> - Asia-Pacífico (Sídney) – `ap-southeast-2`
> - Asia-Pacífico (Tokio) – `ap-northeast-1`
> - Canadá (Centra) – `ca-central-1`
> - China (Pekín): `cn-north-1`
> - China (Ningxia): `cn-northwest-1`
> - Europa (Fráncfort) – `eu-central-1`
> - Europa (Irlanda) – `eu-west-1`
> - Europa (Londres) – `eu-west-2`
> - Europa (París) – `eu-west-3`
> - Europa (Estocolm) – `eu-north-1`
> - América del Sur (São Paulo) – `sa-east-1`

Periodo de retención

Las métricas de S3 Storage Lens se retienen para que pueda ver las tendencias históricas y comparar las diferencias en el uso y la actividad de almacenamiento a lo largo del tiempo. Puede utilizar las métricas de Amazon S3 Storage Lens para consultas de modo que pueda ver las tendencias históricas y comparar las diferencias en el uso y la actividad de almacenamiento a lo largo del tiempo.

Todas las métricas de S3 Storage Lens se retienen durante un periodo de 15 meses. Sin embargo, las métricas solo están disponibles para consultas de una duración específica, que depende de la selección de métricas. Esta duración no se puede modificar. Las métricas gratuitas están disponibles para consultas durante un periodo de 14 días y las métricas avanzadas para consultas durante un periodo de 15 meses.

Categorías de métricas

Dentro de los niveles gratuito y avanzado, las métricas de Lente de almacenamiento de S3 se organizan en categorías que se alinean con los casos de uso clave, como la optimización de costos y la protección de datos. Las métricas gratuitas incluyen métricas de resumen, optimización de costos, protección de datos, gestión de acceso, rendimiento y eventos. Al actualizar a recomendaciones y métricas avanzadas, puede habilitar métricas adicionales de optimización de costes y protección de datos que puede utilizar para reducir aún más los costes de almacenamiento de S3 y garantizar la protección de sus datos. También puede habilitar las métricas de actividad y las métricas de código de estado detalladas que puede usar para mejorar el rendimiento de las cargas de trabajo de las aplicaciones.

La siguiente lista muestra todas las categorías de métricas gratuitas y avanzadas. Para obtener una lista completa de las métricas individuales incluidas en cada categoría, consulte el Metrics glossary (Glosario de métricas).

Métricas de resumen

Las métricas de resumen proporcionan información general sobre el almacenamiento de S3, incluido el total de bytes de almacenamiento y el recuento de objetos.

Métricas de optimización de costos

Las métricas de optimización de costos proporcionan información que puede utilizar para administrar y optimizar los costos de almacenamiento. Por ejemplo, puede identificar los buckets que tengan cargas multipartes incompletas con más de 7 días de antigüedad.

Con las métricas y recomendaciones avanzadas, puede habilitar métricas de optimización de costos avanzadas. Estas métricas incluyen las métricas del recuento de reglas del ciclo de vida de S3 que puede utilizar para obtener los recuentos de reglas del ciclo de vida de S3 de vencimiento y transición por bucket.

Métricas de protección de datos

Las métricas de protección de datos proporcionan información sobre las características de protección de datos, como el cifrado y el control de versiones de S3. Puede utilizar estas métricas para identificar los buckets que no siguen las prácticas recomendadas de protección de datos. Por ejemplo, puede identificar los buckets que no usan el cifrado predeterminado con las claves de AWS Key Management Service (SSE-KMS) o el control de versiones de S3.

Con las métricas y recomendaciones avanzadas, puede habilitar métricas de protección de datos avanzadas. Estas métricas incluyen métricas de recuento de reglas de replicación por bucket.

Métricas de administración de acceso

Las métricas de administración de acceso proporcionan información sobre la propiedad de los objetos de S3. Puede usar estas métricas para ver qué configuración de propiedad de objetos utilizan los buckets.

Métricas de evento

Las métricas de eventos proporcionan información para las notificaciones de eventos de S3. Con las métricas de eventos, puede ver qué buckets tienen configuradas las notificaciones de eventos de S3.

Métricas de desempeño

Las métricas de rendimiento proporcionan información sobre S3 Transfer Acceleration. Con las métricas de rendimiento, puede ver qué buckets tienen habilitada Transfer Acceleration.

Métricas de actividad (avanzadas)

Si actualiza el panel con Métricas y recomendaciones avanzadas, puede habilitar las métricas de actividad. Las métricas de actividad proporcionan detalles sobre cómo se solicita el almacenamiento (por ejemplo, todas las solicitudes, las solicitudes Get, las solicitudes Put), los bytes cargados o descargados y los errores.

Las métricas de actividad a nivel de prefijo se pueden utilizar para determinar qué prefijos se utilizan con poca frecuencia, para así realizar la transición a una clase de almacenamiento más óptima con Ciclo de vida de S3.

Métricas de código de estado detalladas (procedimiento avanzado)

Si actualiza el panel con Métricas y recomendaciones avanzadas, puede habilitar las métricas de código de estado detalladas. Las métricas de código de estado detalladas proporcionan información para los códigos de estado HTTP, como 403 Prohibido y 503 Servicio no disponible, que puede

utilizar para solucionar problemas de acceso o rendimiento. Por ejemplo, puede consultar la métrica 403 Forbidden error count (Recuento de error 403 Forbidden [Prohibido]) para identificar las cargas de trabajo que acceden a los buckets sin haberse aplicado los permisos correctos.

Las métricas detalladas de los códigos de estado a nivel de prefijo se pueden utilizar para comprender mejor los casos de códigos de estado HTTP por prefijo. Por ejemplo, las métricas de recuento de errores 503 permiten identificar los prefijos que reciben solicitudes de limitación durante la ingesta de datos.

Recomendaciones

S3 Storage Lens proporciona recomendaciones automatizadas para ayudarlo a optimizar su almacenamiento. Las recomendaciones se colocan contextualmente junto con las métricas relevantes en el panel de S3 Storage Lens. Los datos históricos no son elegibles para las recomendaciones porque estas son relevantes en lo que sucede en el periodo más reciente. Las recomendaciones aparecen solo cuando son relevantes.

Las recomendaciones de S3 Storage Lens tienen las siguientes formas:

- Sugerencias

  Las sugerencias le alertan sobre tendencias dentro de la actividad y el almacenamiento que podrían indicar una oportunidad de optimización de costes de almacenamiento o prácticas recomendadas de protección de datos. Puede utilizar los temas sugeridos en la Guía para usuarios de Amazon S3 y en el panel de S3 Storage Lens con el fin de obtener más detalles sobre las regiones, los buckets o los prefijos específicos.
- Avisos

  Los avisos son recomendaciones que lo alertan sobre anomalías interesantes dentro de su actividad y almacenamiento durante un periodo que podría requerir más atención o monitorización.
  - Avisos atípicos

    S3 Storage Lens proporciona avisos para métricas que son valores atípicos, según la tendencia reciente de 30 días. El valor atípico se calcula con una puntuación estándar, también conocida como puntuación z. En esta puntuación, la métrica del día actual se resta del promedio de los últimos 30 días de esa métrica. A continuación, la métrica del día actual se divide por la desviación estándar de esa métrica en los últimos 30 días. La puntuación resultante suele estar entre -3 y +3. Este número representa la cantidad de desviaciones estándar que tiene la métrica del día actual de la media.

S3 Storage Lens considera que las métricas con una puntuación >2 o <-2 son valores atípicos porque son superiores o inferiores al 95 % de los datos distribuidos normalmente.

- Avisos sobre cambios significativos

  El aviso sobre cambios significativos se aplica a las métricas que se espera que cambien con menos frecuencia. Por lo tanto, se establece en una sensibilidad mayor que el cálculo de valores atípicos, que normalmente está en el rango de +/- 20 % en comparación con el día, la semana o el mes anterior.

  Abordar los avisos en su almacenamiento y actividad: si recibe un aviso sobre cambios significativos, no es necesariamente un problema. El aviso podría deberse a un cambio anticipado en su almacenamiento. Por ejemplo, es posible que haya agregado recientemente un gran número de objetos nuevos, eliminado un gran número de objetos o realizado cambios planificados similares.

  Si ve un aviso sobre cambios significativos en el panel, tome nota de ello y determine si se puede explicar por circunstancias recientes. Si no es así, utilice el panel de S3 Storage Lens para profundizar en más detalles y comprender las regiones, los buckets o los prefijos específicos que impulsan la fluctuación.

- Recordatorios

  Los recordatorios proporcionan información sobre cómo funciona Amazon S3. Pueden ayudarle a obtener más información sobre las formas de utilizar las características de S3 para reducir los costos de almacenamiento o aplicar las prácticas recomendadas de protección de datos.

Selección de métricas

S3 Storage Lens ofrece dos selecciones de métricas que puede elegir para su panel y exportar: métricas gratuitas y métricas y recomendaciones avanzadas.

- Métricas gratuitas

  S3 Storage Lens ofrece métricas gratuitas para todos los paneles y configuraciones. Las métricas gratuitas contienen métricas que son relevantes para el almacenamiento, como la cantidad de buckets y los objetos de la cuenta. Las métricas gratuitas también incluyen métricas basadas en casos de uso (por ejemplo, métricas de optimización de costos y protección de datos) que puede utilizar para investigar si el almacenamiento está configurado de acuerdo con las prácticas recomendadas de S3. Todas las métricas gratuitas se recopilan diariamente. Los datos están

disponibles para consultas durante 14 días. Para obtener más información sobre las métricas que están disponibles con las métricas gratuitas, consulte Glosario de métricas de Amazon S3 Storage Lens.

- Métricas y recomendaciones avanzadas

  S3 Storage Lens ofrece métricas gratuitas para todos los paneles y configuraciones con la opción de actualizar a métricas y recomendaciones avanzadas. Se aplican cargos adicionales de . Para obtener más información, consulte Precios de Amazon S3.

  Las métricas y recomendaciones avanzadas incluyen todas las métricas de las métricas gratuitas junto con métricas adicionales, como las métricas de protección de datos y optimización de costos avanzadas, las métricas de actividad y las métricas de código de estado detalladas. Las métricas y las recomendaciones avanzadas también proporcionan recomendaciones para ayudarle a optimizar el almacenamiento. Las recomendaciones se colocan contextualmente junto con las métricas relevantes en el panel.

  Las métricas y recomendaciones avanzadas incluyen las siguientes características:

  - Métricas avanzadas: genere métricas adicionales. Para obtener una lista completa de categorías de métricas avanzadas, consulte Categorías de métricas. Para obtener una lista completa de métricas, consulte Glosario de métricas de Amazon S3 Storage Lens.
  - Publicación de Amazon CloudWatch: publica métricas de S3 Storage Lens en CloudWatch para crear una vista unificada del estado operativo en los paneles de CloudWatch. También puede utilizar las características y operaciones de la API de CloudWatch, como alarmas y acciones desencadenadas, cálculos de métricas y detección de anomalías para monitorear y tomar medidas en las métricas de Lente de almacenamiento de S3. Para obtener más información, consulte Monitoreo de métricas de S3 Storage Lens en CloudWatch.
  - Agregación de prefijo: recopile métricas en el nivel de prefijo. Al habilitar la agregación de prefijos, se amplían todas las métricas incluidas en la configuración del panel a nivel de prefijo. Las métricas solo se generan para los prefijos que alcanzan el umbral configurado. Tenga en cuenta que las métricas que son aplicables a nivel de prefijo están disponibles con Agregación de prefijos, excepto las métricas de configuración de nivel de bucket y de recuento de reglas. Las métricas de nivel de prefijo no se publican en CloudWatch.
  - Agregación de grupos de Lente de almacenamiento: recopila métricas a nivel de grupo de Lente de almacenamiento. Tras habilitar Métricas y recomendaciones avanzadas y Agregación de grupos de Lente de almacenamiento, puede especificar qué grupos de Lente de almacenamiento desea incluir o excluir del panel de Lente de almacenamiento. Debe

especificarse al menos un grupo de Lente de almacenamiento. Los grupos de Lente de almacenamiento que se especifiquen también deben encontrarse en la región de origen designada de la cuenta del panel. Las métricas de nivel de grupo de Storage Lens no se publican en CloudWatch.

Todas las métricas avanzadas se recopilan diariamente. Los datos están disponibles para consultas durante 15 meses. Para obtener más información acerca de las métricas de almacenamiento que se agregan por Lente de almacenamiento de S3, consulte Glosario de métricas de Amazon S3 Storage Lens.

> **Note**
>
> Las recomendaciones solo están disponibles cuando utiliza el panel de S3 Storage Lens en la consola de Amazon S3.

S3 Storage Lens y AWS Organizations

AWS Organizations es un Servicio de AWS que le ayuda a agregar todas las Cuentas de AWS bajo una jerarquía de organización. Amazon S3 Storage Lens trabaja con AWS Organizations para proporcionar una vista única de la actividad y almacenamiento de objetos en todo el almacenamiento de Amazon S3.

Para obtener más información, consulte Uso de Amazon S3 Storage Lens con AWS Organizations.

- Acceso de confianza

  Con la cuenta de administración de la organización, debe habilitar el acceso de confianza para Lente de almacenamiento de S3 a fin de agregar métricas de almacenamiento y datos de uso para todas las cuentas de miembros de la organización. A continuación, puede crear paneles o exportaciones para la organización mediante la cuenta de administración o brindando acceso de administrador delegado a otras cuentas de la organización.

  Puede deshabilitar el acceso de confianza para S3 Storage Lens en cualquier momento, lo que impide que S3 Storage Lens agregue métricas para su organización.

- Administrador delegado

  Puede crear paneles y métricas para Lente de almacenamiento de S3 destinados a la organización mediante la cuenta de administración de AWS Organizations u otorgando acceso de administrador

delegado a otras cuentas de la organización. Puede anular el registro de los administradores delegados en cualquier momento. La anulación del registro de un administrador delegado también impide automáticamente que todos los paneles de nivel de organización que haya creado ese administrador delegado agreguen nuevas métricas de almacenamiento.

Para obtener más información, consulte Amazon S3 Storage Lens y AWS Organizations en la Guía del usuario de AWS Organizations.

Roles vinculados a servicios de Amazon S3 Storage Lens

Junto con el acceso de confianza de AWS Organizations, Amazon S3 Storage Lens utiliza roles vinculados a servicios de AWS Identity and Access Management (IAM). Un rol vinculado a un servicio es un tipo único de rol de IAM que está vinculado directamente a S3 Storage Lens. Los roles vinculados a servicios están predefinidos por S3 Storage Lens e incluyen todos los permisos necesarios para recopilar las métricas diarias de actividad y almacenamiento de las cuentas de miembros de su organización.

Para obtener más información, consulte Uso de roles vinculados a servicios para Amazon S3 Storage Lens.

Uso de Amazon S3 Storage Lens con AWS Organizations

La Lente de almacenamiento de Amazon S3 es una función de análisis de almacenamiento en la nube que puede utilizar para obtener visibilidad en toda la organización sobre el uso y la actividad del almacenamiento de objetos. Puede utilizar las métricas de S3 Storage Lens para generar información resumida, como averiguar cuánto almacenamiento tiene en toda la organización o cuáles son los buckets y los prefijos de crecimiento más rápido. También puede utilizar las métricas de Amazon S3 Storage Lens para identificar oportunidades de optimización de costos, implementar las prácticas recomendadas de protección y seguridad de los datos y mejorar el rendimiento de las cargas de trabajo de las aplicaciones. Por ejemplo, puede identificar los buckets que no tienen reglas del ciclo de vida de S3 para que hagan vencer las cargas multipartes incompletas que tengan más de 7 días de antigüedad. También puede identificar los buckets que no siguen las prácticas recomendadas de protección de datos, como el uso de la Replicación de S3 o el control de versiones de S3. Lente de almacenamiento de S3 analiza también las métricas para ofrecer recomendaciones contextuales que puede usar para optimizar los costos de almacenamiento y aplicar las prácticas recomendadas para proteger los datos.

Puede utilizar Amazon S3 Storage Lens para recopilar métricas de almacenamiento y datos de uso de todas las Cuentas de AWS que forman parte de la jerarquía de AWS Organizations. Para ello,

debe utilizar AWS Organizations y habilitar el acceso de confianza de Lente de almacenamiento de S3 con la cuenta de administración de AWS Organizations.

Después de habilitar el acceso de confianza, puede agregar acceso de administrador delegado a las cuentas de la organización. Estas cuentas pueden crear configuraciones y paneles de S3 Storage Lens que recopilen métricas de almacenamiento y datos de usuario de toda la organización.

Para obtener más información acerca de cómo habilitar el acceso de confianza, consulte Amazon S3 Storage Lens y AWS Organizations en la Guía del usuario de AWS Organizations.

Temas

Habilitación del acceso de confianza para S3 Storage Lens

Si habilita el acceso de confianza, permitirá que la Lente de almacenamiento de Amazon S3 tenga acceso a la jerarquía, la pertenencia y la estructura de AWS Organizations a través de las operaciones de la API de AWS Organizations. Lente de almacenamiento de S3 se convierte entonces en un servicio de confianza para toda la estructura de la organización.

Cada vez que se crea una configuración de panel, Lente de almacenamiento de S3 crea roles vinculados a servicios en las cuentas de administración o administrador delegadas de la organización. El rol vinculado a servicios concede permiso a Lente de almacenamiento de S3 para hacer lo siguiente:

- Describir organizaciones
- Enumerar cuentas
- Comprobar una lista de accesos de Servicio de AWS para las organizaciones
- Obtener administradores delegados para las organizaciones

Así, Lente de almacenamiento de S3 puede asegurarse de tener acceso para recopilar las métricas entre cuentas para las cuentas en las organizaciones. Para obtener más información, consulte Uso de roles vinculados a servicios para Amazon S3 Storage Lens.

Después de habilitar el acceso de confianza, puede asignar acceso de administrador delegado a cuentas de la organización. Cuando una cuenta se marca como administrador delegado para un servicio, la cuenta recibe autorización para acceder a todas las operaciones de la API de organización de solo lectura. Este acceso proporciona visibilidad de administrador delegado a los miembros y las estructuras de la organización para que también puedan crear paneles de Lente de almacenamiento de S3.

> **Note**
>
> Solo la cuenta de administración puede habilitar el acceso de confianza para Amazon S3 Storage Lens.

Deshabilitación del acceso de confianza para S3 Storage Lens

Con la deshabilitación del acceso de confianza, limita a S3 Storage Lens a trabajar solo en el nivel de cuenta. Además, cada titular de la cuenta solo puede ver la información de Lente de almacenamiento de S3 para el alcance de la cuenta y no toda la organización. Los paneles que requieren acceso de confianza ya no se actualizarán, pero retendrán los datos históricos para el periodo en el que los datos están disponibles para consultas.

> **Note**
>
> - La desactivación del acceso de confianza para Lente de almacenamiento de S3 impide automáticamente que todos los paneles en el nivel de organización recopilen y agreguen métricas de almacenamiento.
> - Sus cuentas de administración y administrador delegado podrán ver los datos históricos de los paneles de nivel de organización existentes durante el periodo en el que los datos están disponibles para consultas.

Registro de un administrador delegado para S3 Storage Lens

Puede crear paneles de nivel de organización con la cuenta de administración o las cuentas de administrador delegado de la organización. Las cuentas de administrador delegado permiten a otras cuentas, además de su cuenta de administración, crear paneles de nivel de organización. Solo la cuenta de administración de una organización puede registrar y anular el registro de otras cuentas como administradores delegados para la organización.

Para registrar un administrador delegado mediante la consola de Amazon S3, consulte Registro de administradores delegados para S3 Storage Lens.

También puede registrar a un administrador delegado con la API de REST de AWS Organizations, la AWS CLI o los SDK de la cuenta de administración. Para obtener más información, consulte RegisterDelegatedAdministrator en la Referencia de la API de AWS Organizations.

> **Note**
>
> Antes de poder designar a un administrador delegado con la API de REST de AWS Organizations, la AWS CLI o los SDK, debe llamar a la operación EnableAWSOrganizationsAccess.

Anulación del registro de un administrador delegado para S3 Storage Lens

También puede anular el registro de una cuenta de administrador delegado. Las cuentas de administrador delegado permiten a otras cuentas, además de su cuenta de administración, crear paneles de nivel de organización. Solo la cuenta de administración de una organización puede anular el registro de cuentas como administradores delegados para la organización.

Para anular el registro de un administrador delegado mediante la consola de S3, consulte Anular el registro de administradores delegados para S3 Storage Lens.

También puede anular el registro de un administrador delegado con la API de REST de AWS Organizations, la AWS CLI o los SDK de la cuenta de administración. Para obtener más información, consulte DeregisterDelegatedAdministrator en la Referencia de la API de AWS Organizations.

Note

- La anulación del registro de un administrador delegado también impide automáticamente que todos los paneles de nivel de organización que haya creado ese administrador delegado agreguen nuevas métricas de almacenamiento.
- El administrador delegado sin registro todavía podrá ver los datos históricos de los paneles que ha creado mientras los datos estén disponibles para consultas.

Permisos de Amazon S3 Storage Lens

Amazon S3 Storage Lens requiere nuevos permisos en AWS Identity and Access Management (IAM) para autorizar el acceso a las acciones de S3 Storage Lens. Para conceder estos permisos, puede utilizar una política de IAM basada en identidades. Puede asociar esta política a usuarios, grupos o roles de IAM para concederles permisos. Estos permisos pueden incluir la capacidad de habilitar o deshabilitar Lente de almacenamiento de S3, o de acceder a cualquier panel o configuración de Lente de almacenamiento de S3.

El rol o usuario de IAM debe pertenecer a la cuenta creadora o propietaria del panel o la configuración, a menos que se cumplan las dos condiciones siguientes:

- Su cuenta es miembro de AWS Organizations.
- Se le concedió permiso para crear paneles de nivel de organización por su cuenta de administración como administrador delegado.

Note

- No puede utilizar las credenciales de usuario raíz de la cuenta para ver los paneles de la Lente de almacenamiento de Amazon S3. Para acceder a los paneles de Lente de almacenamiento de S3, debe conceder los permisos de IAM necesarios a un usuario de IAM nuevo o existente. A continuación, inicie sesión con esas credenciales de usuario para acceder a los paneles de S3 Storage Lens. Para más información, consulte Prácticas recomendadas de seguridad en IAM en la Guía del usuario de IAM.
- El uso de S3 Storage Lens en la consola de Amazon S3 puede requerir varios permisos. Por ejemplo, para editar un panel en la consola, necesita los siguientes permisos:

- `s3:ListStorageLensConfigurations`
- `s3:GetStorageLensConfiguration`
- `s3:PutStorageLensConfiguration`

Temas

- Configuración de permisos de cuenta para usar S3 Storage Lens
- Configuración de permisos de cuenta para usar grupos de S3 Storage Lens
- Configuración de permisos para utilizar S3 Storage Lens con AWS Organizations

Configuración de permisos de cuenta para usar S3 Storage Lens

Para crear y administrar los paneles de Lente de almacenamiento de S3 y las configuraciones de los paneles de Lente de almacenamiento, debe tener los siguientes permisos, en función de las acciones que desee realizar:

Permisos de IAM relacionados con Amazon S3 Storage Lens

Acción	Permisos de IAM
Cree o actualice un panel de S3 Storage Lens en la consola de Amazon S3.	`s3:ListStorageLensConfigurations` `s3:GetStorageLensConfiguration` `s3:GetStorageLensConfigurationTagging` `s3:PutStorageLensConfiguration` `s3:PutStorageLensConfigurationTagging`
Obtenga las etiquetas de un panel de Lente de almacenamiento de S3 en la consola de Amazon S3.	`s3:ListStorageLensConfigurations` `s3:GetStorageLensConfigurationTagging`
Vea un panel de S3 Storage Lens en la consola de Amazon S3.	`s3:ListStorageLensConfigurations`

Acción	Permisos de IAM
	`s3:GetStorageLensConfiguration` `s3:GetStorageLensDashboard`
Eliminar un panel de S3 Storage Lens en la consola de Amazon S3.	`s3:ListStorageLensConfigurations` `s3:GetStorageLensConfiguration` `s3:DeleteStorageLensConfiguration`
Cree o actualice una configuración de Lente de almacenamiento de S3 mediante la AWS CLI o un SDK de AWS.	`s3:PutStorageLensConfiguration` `s3:PutStorageLensConfigurationTagging`
Obtenga las etiquetas de una configuración de Lente de almacenamiento de S3 usando la AWS CLI o un SDK de AWS.	`s3:GetStorageLensConfigurationTagging`
Vea una configuración de Lente de almacenamiento de S3 mediante la AWS CLI o un SDK de AWS.	`s3:GetStorageLensConfiguration`
Elimine una configuración de Lente de almacenamiento de S3 mediante la AWS CLI o el SDK de AWS.	`s3:DeleteStorageLensConfiguration`

Note

- Las vistas del panel de Lente de almacenamiento de Amazon S3 se registran en CloudTrail con el nombre de evento `GetStorageLensDashboardDataInternal`.
- Puede utilizar etiquetas de recursos en una política del IAM para administrar permisos.
- Un usuario o rol de IAM con estos permisos puede ver métricas de buckets y prefijos de los que es posible que no tenga permiso directo para leer o mostrar objetos.

- En el caso de los paneles de Lente de almacenamiento de S3 con las métricas a nivel de prefijo habilitadas, si una ruta de prefijo seleccionada coincide con una clave de objeto, es posible que el panel muestre la clave de objeto como otro prefijo.
- Para las exportaciones de métricas, que se almacenan en un bucket de la cuenta, los permisos se otorgan mediante el permiso `s3:GetObject` existente en la política de IAM. Del mismo modo, para una entidad de AWS Organizations, la cuenta de administración o de administrador delegado de la organización puede utilizar políticas de IAM para administrar los permisos de acceso para las configuraciones y los paneles en el nivel de la organización.

Configuración de permisos de cuenta para usar grupos de S3 Storage Lens

Puede usar los grupos de Lente de almacenamiento de S3 para comprender la distribución de su almacenamiento dentro de los buckets en función del prefijo, el sufijo, o la etiqueta, el tamaño o la antigüedad de los objetos. Puede asociar grupos de Lente de almacenamiento a sus paneles para ver sus métricas agregadas.

Para trabajar con los grupos de Lente de almacenamiento, necesita determinados permisos. Para obtener más información, consulte the section called “Permisos de grupos de Storage Lens”.

Configuración de permisos para utilizar S3 Storage Lens con AWS Organizations

Puede utilizar Amazon S3 Storage Lens para recopilar métricas de almacenamiento y datos de uso de todas las cuentas que forman parte de la jerarquía de AWS Organizations. Las siguientes son las acciones y permisos relacionados con el uso de S3 Storage Lens con Organizations.

AWS OrganizationsPermisos de IAM relacionados con para el uso de S3 Storage Lens

Acción	Permisos de IAM
Habilite el acceso de confianza para S3 Storage Lens para su organización.	`organizations:EnableAWSServiceAccess`
Desactive el acceso de confianza para Lente de almacenamiento de S3 para la organización.	`organizations:DisableAWSServiceAccess`

Acción	Permisos de IAM
Registre a un administrador delegado a fin de crear paneles o configuraciones de S3 Storage Lens para su organización.	`organizations:RegisterDelegatedAdministrator`
Anule el registro de un administrador delegado para que no pueda crear paneles o configuraciones de Lente de almacenamiento de S3 para la organización.	`organizations:DeregisterDelegatedAdministrator`
Permisos adicionales a fin de crear configuraciones de Lente de almacenamiento de S3 para toda la organización.	`organizations:DescribeOrganization` `organizations:ListAccounts` `organizations:ListAWSServiceAccessForOrganization` `organizations:ListDelegatedAdministrators` `iam:CreateServiceLinkedRole`

Visualización de métricas con Lente de almacenamiento de Amazon S3

Lente de almacenamiento de S3 agrega las métricas y muestra la información en la sección Account snapshot (Instantánea de la cuenta) en la página Buckets de la consola de Amazon S3. Lente de almacenamiento de S3 también proporciona un panel interactivo que puede utilizar para visualizar información y tendencias, marcar valores atípicos y recibir recomendaciones para optimizar los costos de almacenamiento y aplicar las prácticas recomendadas de protección de datos. El panel tiene opciones de desglose para generar y visualizar información en el nivel de la organización, la cuenta, la Región de AWS, la clase de almacenamiento, el bucket, el prefijo o el grupo de Storage Lens. También puede enviar una exportación de métricas diaria en formato CSV o Parquet a un bucket de S3.

De forma predeterminada, todos los paneles están configurados con métricas gratuitas, que incluyen métricas que puede utilizar para comprender el uso y la actividad de almacenamiento de S3,

optimizar los costos de almacenamiento e implementar las prácticas recomendadas de protección de datos y administración de acceso. Las métricas gratuitas se agregan hasta el nivel de bucket. Con métricas gratuitas, los datos están disponibles para consultas hasta un máximo de 14 días.

Las métricas y recomendaciones avanzadas incluyen las siguientes funciones adicionales que puede utilizar para obtener más información sobre el uso y la actividad del almacenamiento, así como las prácticas recomendadas para optimizarlo:

- Recomendaciones contextuales (disponibles solo en el panel)
- Métricas avanzadas (incluidas las métricas de actividad agregadas por bucket)
- Agregación de prefijos
- Agregación de grupos de Storage Lens
- Agregación de grupos de Storage Lens
- Publicación de Amazon CloudWatch

Los datos de métricas avanzadas están disponibles para consultas durante 15 meses. Existen cargos adicionales por usar S3 Storage Lens con métricas avanzadas. Para obtener más información, consulte Precios de Amazon S3. Para obtener más información sobre las métricas gratuitas y avanzadas, consulte Selección de métricas.

Temas

- Visualización de las métricas de S3 Storage Lens en los paneles
- Visualización de las métricas de Amazon S3 Storage Lens mediante una exportación de datos
- Monitoreo de métricas de S3 Storage Lens en CloudWatch

Visualización de las métricas de S3 Storage Lens en los paneles

En la consola de Amazon S3, S3 Storage Lens proporciona un panel predeterminado interactivo que puede utilizar para visualizar información y tendencias de los datos. Puede utilizar este panel para visualizar la información y las tendencias, marcar los valores atípicos y obtener las recomendaciones necesarias para optimizar los costes de almacenamiento y aplicar las prácticas recomendadas de protección de datos. El panel tiene opciones de desglose para generar información en el nivel de la cuenta, el bucket, la Región de AWS, el prefijo o el grupo de Lente de almacenamiento. Si ha habilitado S3 Storage Lens para que funcione con AWS Organizations, también puede generar información en el nivel de la organización (como datos para todas las cuentas que forman parte de

su jerarquía de AWS Organizations). El panel siempre se carga en la fecha más reciente que tiene métricas disponibles.

El panel predeterminado de S3 Storage Lens en la consola se denomina default-account-dashboard. Amazon S3 preconfigura este panel para visualizar la información resumida y las tendencias de toda la cuenta y las actualiza a diario en la consola de S3. No puede modificar el alcance de la configuración del panel predeterminado, pero puede actualizar la selección de métricas gratuitas a las métricas y recomendaciones avanzadas pagadas. Con métricas y recomendaciones avanzadas, puede acceder a métricas y características adicionales. Estas características incluyen las categorías de métricas avanzadas, la agregación de nivel de prefijo, las recomendaciones contextuales y la publicación de Amazon CloudWatch.

Puede desactivar el panel predeterminado, pero no puede eliminarlo. Si desactiva el panel predeterminado, ya no se actualizará. Ya no recibirá ninguna métrica diaria nueva en S3 Storage Lens ni en la sección Instantánea de cuenta en la página de Buckets. Aún podrá ver los datos históricos en el panel predeterminado hasta el periodo de 14 días para que venzan las consultas de datos. Este periodo es de 15 meses si ha habilitado las métricas y recomendaciones avanzadas. Para acceder a estos datos, puede volver a habilitar el panel predeterminado dentro del periodo de vencimiento.

Puede crear paneles de S3 Storage Lens adicionales y segmentarlos por Regiones de AWS, buckets de S3 o cuentas. También puede organizar sus paneles por organización si ha habilitado Storage Lens para trabajar con AWS Organizations. Cuando crea o edita un panel de Lente de almacenamiento de S3, define el alcance del panel y la selección de métricas.

Puede desactivar o eliminar cualquier panel adicional que cree.

- Si deshabilita el panel, ya no se actualizará y ya no recibirá métricas diarias nuevas. Aún podrá ver los datos históricos para las métricas gratuitas hasta el periodo de vencimiento de 14 días. Si ha habilitado las métricas y recomendaciones avanzadas para ese panel, este periodo es de 15 meses. Para acceder a estos datos, puede volver a habilitar el panel dentro del periodo de vencimiento.
- Si elimina el panel, perderá todas las opciones de configuración del panel. Ya no recibirá métricas diarias nuevas y también perderá el acceso a los datos históricos asociados a ese panel. Si desea acceder a los datos históricos de un panel eliminado, debe crear otro panel con el mismo nombre en la misma región principal.

Temas

- Visualización de un panel de Amazon S3 Storage Lens
- Comprensión del panel de S3 Storage Lens

Visualización de un panel de Amazon S3 Storage Lens

El siguiente procedimiento muestra cómo ver un panel de Lente de almacenamiento de S3 en la consola de S3. Para ver tutoriales basados en casos de uso que muestran cómo utilizar el panel para optimizar los costes, implementar prácticas recomendadas y mejorar el rendimiento de las aplicaciones que acceden a los buckets de S3, consulte Casos de uso de métricas de lente de almacenamiento de Amazon S3.

> **Note**
>
> No puede utilizar las credenciales de usuario raíz de la cuenta para ver los paneles de la Lente de almacenamiento de Amazon S3. Para acceder a los paneles de Lente de almacenamiento de S3, debe conceder los permisos de AWS Identity and Access Management necesarios a un usuario de IAM nuevo o existente. A continuación, inicie sesión con esas credenciales de usuario para acceder a los paneles de S3 Storage Lens. Para obtener más información, consulte Permisos de Amazon S3 Storage Lens y Prácticas recomendadas de seguridad de IAM en la Guía del usuario de IAM.

1. Inicie sesión en la AWS Management Console y abra la consola de Amazon S3 en https://console.aws.amazon.com/s3/.
2. En el panel de navegación izquierdo, elija Storage Lens, Dashboards (Paneles).
3. En la lista Dashboards (Paneles), elija el panel que desea ver.

   El panel se abre en Lente de almacenamiento de S3. La sección Snapshot for date (Instantánea para fecha) muestra la fecha más reciente para la que Lente de almacenamiento de S3 ha recopilado métricas. El panel siempre se carga en la fecha más reciente que tiene métricas disponibles.
4. (Opcional) Para cambiar la fecha del panel de Lente de almacenamiento de S3, en el selector de fechas de la esquina superior derecha, elija una nueva fecha.
5. (Opcional) Para aplicar filtros temporales para limitar aún más el alcance de los datos del panel, haga lo siguiente:

   a. Expanda la sección Filtros.

b. Para filtrar por cuentas específicas, Regiones de AWS, clases de almacenamiento, buckets, prefijos o grupos de Storage Lens específicos, elija las opciones por las que desea filtrar.

> **Note**
>
> El filtro Prefijos y el filtro Grupos de Storage Lens no se pueden aplicar al mismo tiempo.

c. Para actualizar un filtro, elija Apply (Aplicar).

d. Para eliminar un filtro, haga clic en la X situada junto al filtro.

6. En cualquier sección del panel de Lente de almacenamiento de S3, para ver los datos de una métrica específica, en Metric (Métrica), seleccione el nombre de la métrica.

7. En cualquier gráfico o visualización del panel de S3 Storage Lens, puede profundizar en los niveles de agregación más profundos mediante las pestañas Cuentas, Regiones de AWS, Clases de almacenamiento, Buckets, Prefijos o Grupos de Storage Lens. Para ver un ejemplo, consulte Descubra buckets en frío de Amazon S3.

Comprensión del panel de S3 Storage Lens

El panel de Lente de almacenamiento de S3 tiene una pestaña principal Overview (Información general) y hasta cinco pestañas adicionales que representan cada nivel de agregación:

- Cuentas
- Regiones de AWS
- Clases de almacenamiento
- Buckets
- Prefijos
- Grupos de Storage Lens

En la pestaña Overview (Información general), los datos del panel se agregan en tres secciones diferentes: Snapshot for date (Instantánea para fecha), Trends and distributions (Tendencias y distribuciones) y Top N overview (Información general de N principales).

Para obtener más información acerca del panel de Lente de almacenamiento de S3, consulte las siguientes secciones.

Instantánea

La sección Snapshot for date (Instantánea para fecha) muestra las métricas de resumen que Lente de almacenamiento de S3 ha agregado para la fecha seleccionada. En estas métricas de resumen se incluyen las siguientes:

- Almacenamiento total: la cantidad total de almacenamiento utilizada en bytes.
- Recuento de objetos: el número total de objetos en su Cuenta de AWS.
- Promedio de tamaño de objetos: el promedio del tamaño de los objetos.
- Buckets activos: el número total de buckets activos en uso activo con almacenamiento mayor que 0 bytes en su cuenta.
- Cuentas: el número de cuentas cuyo almacenamiento está en el alcance. Este valor será 1, a menos que utilice AWS Organizations y Lente de almacenamiento de S3 tenga acceso de confianza con un rol vinculado a servicios válido. Para obtener más información, consulte Uso de roles vinculados a servicios para Amazon S3 Storage Lens.
- Buckets: el número total de buckets en su cuenta.

Datos de métricas

Para cada métrica que aparece en la instantánea, puede ver los siguientes datos:

- Nombre de métrica: el nombre de la métrica.
- Categoría de métrica: la categoría en la que está organizada la métrica.
- Total para la fecha: el recuento total para la fecha seleccionada.
- Porcentaje de cambio: el cambio porcentual con respecto a la fecha de la última instantánea.
- Tendencia de 30 días: una línea de tendencia que muestra los cambios de la métrica durante un periodo de 30 días.
- Recomendación: una recomendación contextual basada en los datos que se proporcionan en la instantánea. Las recomendaciones están disponibles con métricas y recomendaciones avanzadas. Para obtener más información, consulte Recomendaciones.

Categorías de métricas

Si lo desea, puede actualizar la sección Snapshot for date (Instantánea para fecha) del panel para mostrar las métricas para otras categorías. Si quiere ver datos de las instantáneas para métricas adicionales, puede elegir entre las siguientes Metrics categories (Categorías de métricas):

- Optimización de costos
- Protección de los datos
- Actividad (disponible con métricas avanzadas)
- Administración de accesos
- Rendimiento
- Eventos

La sección Snapshot for date (Instantánea para fecha) muestra solo una selección de métricas para cada categoría. Para ver todas las métricas de una categoría específica, elija la métrica en las secciones Trends and distributions (Tendencias y distribuciones) o Top N overview (Información general de N principales). Para obtener más información sobre categorías de métricas, consulte Categorías de métricas. Para obtener una lista completa de las métricas de Lente de almacenamiento de S3, consulte Glosario de métricas de Amazon S3 Storage Lens.

Tendencias y distribuciones

La segunda sección de la pestaña Overview (Información general) es Trends and distributions (Tendencias y distribuciones). En la sección Trends and distributions (Tendencias y distribuciones), puede elegir dos métricas para compararlas en un intervalo de fechas que defina. La sección Trends and distributions (Tendencias y distribuciones) muestra la relación entre dos métricas a lo largo del tiempo. En esta sección se muestran gráficos que puede usar para ver la distribución de Storage class (Clase de almacenamiento) y Region (Región) entre dos tendencias que está rastreando. Si lo desea, puede profundizar en un punto de datos de uno de los gráficos para realizar un análisis más profundo.

Para ver un tutorial que utiliza la sección Trends and distributions (Tendencias y distribuciones), consulte Identificar los buckets que no utilizan el cifrado del lado del servidor con AWS KMS para el cifrado predeterminado (SSE-KMS).

Descripción general de las N principales

La tercera sección del panel S3 Storage Lens es la descripción general de las N principales (ordenadas en orden ascendente o descendente). Esta sección le permite ver las métricas seleccionadas en el número de cuentas principales, Regiones de AWS, buckets, prefijos o grupos de Storage Lens. Si ha habilitado S3 Storage Lens para que funcione con AWS Organizations, también puede ver las métricas seleccionadas en toda la organización.

Para ver un tutorial que utilice la sección Top N overview (Información general de N principales), consulte Identifique sus buckets más grandes de S3.

Profundizar y analizar por opciones

Para proporcionar una experiencia fluida de análisis, el panel de S3 Storage Lens proporciona un menú de acción, que aparece cuando elige cualquier valor de gráfico. Para usar este menú, elija cualquier valor de gráfico para ver los valores de métrica asociados y, a continuación, elija entre las dos opciones del cuadro que aparece:

- La acción de desglose aplica el valor seleccionado como filtro en todas las pestañas del panel. A continuación, puede desglosar ese valor para un análisis más profundo.
- La acción de Analizar por le lleva a la pestaña Dimensión seleccionada y aplica ese valor de pestaña como un filtro. Estas pestañas incluyen Cuentas, Regiones de AWS, Clases de almacenamiento, Buckets, Prefijos (para los paneles que tienen habilitadas las opciones Métricas avanzadas y Agregación de prefijos) y los Grupos de Storage Lens (para los paneles que tienen habilitadas las opciones Métricas avanzadas y Agregación de grupos de Storage Lens). Con Analizar por, puede ver los datos en el contexto de la nueva dimensión para un análisis más profundo.

Es posible que las acciones de Desglose descendente y Analizar por estén deshabilitadas si el resultado es ilógico o no tuviera valores. Las acciones de Desglose descendente y Analizar por aplican filtros sobre los filtros existentes en todas las pestañas del panel. También puede eliminar los filtros según sea necesario.

Pestañas

Las pestañas de nivel de dimensión proporcionan una vista detallada de todos los valores dentro de una dimensión en particular. Por ejemplo, en la pestaña Regiones de AWS se muestran las métricas de todas las Regiones de AWS, y en la pestaña Bucket se muestran las métricas de todos los buckets. Cada pestaña de dimensión contiene un diseño idéntico que consta de cuatro secciones:

- Un gráfico de tendencias que muestra los elementos N principales dentro de la dimensión durante los últimos 30 días para la métrica seleccionada. De forma predeterminada, este gráfico muestra los 10 elementos principales, pero puede reducirlo a al menos 3 elementos o aumentarlo a 50 elementos.

- Un gráfico de histograma que muestra un gráfico de barras verticales para la fecha y la métrica seleccionadas. Si tiene un número grande de elementos para mostrar en este gráfico, es posible que se deba desplazar horizontalmente.
- Un gráfico de análisis de burbujas que representa todos los elementos de la dimensión. Este gráfico representa la primera métrica en el eje x y la segunda métrica en el eje y. La tercera métrica está representada por el tamaño de la burbuja.
- Vista de cuadrícula de métricas que contiene cada elemento de la dimensión enumerada en filas. Las columnas representan cada métrica disponible, organizadas en pestañas de categorías de métricas para facilitar la navegación.

Visualización de las métricas de Amazon S3 Storage Lens mediante una exportación de datos

Las métricas de la Lente de almacenamiento de Amazon S3 se generan diariamente en archivos de exportación de métricas con formato CSV o Apache Parquet y se colocan en un bucket de S3 en la cuenta. Desde allí, puede incorporar las métricas exportadas en las herramientas de análisis de su elección, como Amazon QuickSight y Amazon Athena, donde puede analizar el uso del almacenamiento y las tendencias de actividad.

Temas

Uso de una AWS KMS key para cifrar las exportaciones de métricas

Si desea conceder permiso a la Lente de almacenamiento de Amazon S3 para cifrar las exportaciones de métricas con una clave administrada por el cliente, debe utilizar una política de claves. Para actualizar la política de claves con el fin de poder usar una clave KMS para cifrar las exportaciones de métricas de Lente de almacenamiento de S3, siga estos pasos.

Para conceder permisos a Lente de almacenamiento de S3 para cifrar datos mediante la clave KMS

1. Inicie sesión en la AWS Management Console con la Cuenta de AWS que es propietaria de la clave administrada por el cliente.
2. Abra la consola de AWS KMS en https://console.aws.amazon.com/kms.

3. Para cambiar la Región de AWS, utilice el Selector de regiones ubicado en la esquina superior derecha de la página.
4. En el panel de navegación izquierdo, elija Customer managed keys (Claves administrada por el cliente).
5. En Claves administradas por el ciente, elija la clave que desea usar para cifrar las exportaciones de métricas. AWS KMS keys son específicas de la región y deben estar en la misma región que el bucket de S3 de destino de la exportaicón de métricas.
6. En Política de claves, seleccione Cambiar a la vista de política.
7. Para actualizar la política de claves, elija Editar.
8. En Editar política de claves, agregue la siguiente política de claves a la política de claves existente. Para utilizar esta política, sustituya *`user input placeholders`* por la información.

```
{
    "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. Elija Guardar cambios.

Para obtener más información acerca de cómo crear claves administradas por el cliente y de cómo usar las políticas de claves, consulte los siguientes temas en la AWS Key Management Service Guía de desarrolladores:

- Introducción
- Uso de políticas de claves en AWS KMS

Puede usar también la operación de la API de política de claves PUT de AWS KMS (PutKeyPolicy) para copiar la política de claves en las claves administradas por el cliente que desea utilizar para cifrar las exportaciones de métricas con la API de REST, la AWS CLI y los SDK.

¿Qué es un manifiesto de exportación de S3 Storage Lens?

Dada la gran cantidad de datos agregados, una exportación diaria de métricas de S3 Storage Lens se puede dividir en varios archivos. El archivo de manifiesto `manifest.json` describe dónde se encuentran los archivos de exportación de métricas para ese día. Cada vez que se entrega una exportación nueva, se acompaña con un manifiesto nuevo. Cada manifiesto incluido en el archivo `manifest.json` proporciona metadatos y otra información básica sobre la exportación.

La información del manifiesto incluye las siguientes propiedades:

- `sourceAccountId` – El ID de cuenta del propietario de la configuración.
- `configId` – Un identificador único para el panel.
- `destinationBucket` – El nombre de recurso de Amazon (ARN) del bucket de destino en el que se coloca la exportación de métricas.
- `reportVersion` – La versión de la exportación.
- `reportDate` – La fecha del informe.
- `reportFormat` – El formato del informe.
- `reportSchema` – El esquema del informe.
- `reportFiles` – La lista real de los archivos de informe de exportación que se encuentran en el bucket de destino.

A continuación se incluye un ejemplo de un manifiesto en un archivo `manifest.json` para una exportación con formato 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"
        }
      ]
}
```

A continuación se incluye un ejemplo de un manifiesto en un archivo `manifest.json` para una exportación con formato 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"
      }
}
```

Puede configurar la exportación de métricas para que se genere como parte de la configuración del panel en la consola de Amazon S3 o mediante el uso de la API de REST de Amazon S3, la AWS CLI y los SDK.

Comprensión del esquema de exportación de Amazon S3 Storage Lens

La tabla siguiente contiene el esquema de la exportación de métricas de S3 Storage Lens.

Nombre de atributo	Tipo de datos	Nombre de la columna	Descripción
`VersionNumber`	Cadena	`version_number`	La versión de las métricas de S3 Storage Lens en uso.
`ConfigurationId`	Cadena	`configuration_id`	El `configuration_id` de la configuración de Lente de almacenamiento de S3.
`ReportDate`	Cadena	`report_date`	La fecha en que se realizó el seguimiento de las métricas.
`AwsAccountNumber`	Cadena	`aws_account_number`	Su número de Cuenta de AWS.
`AwsRegion`	Cadena	`aws_region`	La Región de AWS para la que se realiza el seguimiento de las métricas.
`StorageClass`	Cadena	`storage_class`	La clase de almacenamiento del bucket en cuestión.
`RecordType`	ENUM	`record_type`	El tipo de artefacto que se notifica (CUENTA, BUCKET o PREFIJO).
`RecordValue`	Cadena	`record_value`	El valor del artefacto `RecordType` .

Nombre de atributo	Tipo de datos	Nombre de la columna	Descripción
			Note `record_value` está codificado en la URL.
`BucketName`	Cadena	`bucket_name`	El nombre del bucket que se informa.
`MetricName`	Cadena	`metric_name`	El nombre de la métrica que se informa.
`MetricValue`	Long	`metric_value`	El valor de la métrica que se informa.

Ejemplo de una exportación de métricas de S3 Storage Lens

A continuación se muestra un ejemplo de una exportación de métricas de S3 Storage Lens basada en este esquema.

Note

Para identificar las métricas de los grupos de Lente de almacenamiento, busque los valores `STORAGE_LENS_GROUP_ACCOUNT` o `STORAGE_LENS_GROUP_BUCKET` en la columna `record_type`. La columna `record_value` mostrará el nombre de recurso de Amazon (ARN) del grupo de Lente de almacenamiento, por ejemplo, `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%2Fus	cloudtrail-log-sf	StorageBytes	29996800
V_1	sample-cmh-exclude	11/3/2020	546264889236	us-west-1	STANDARD	PREFIX	AWSLogs%2F546264889236%2FCloudTrail%2Fus	cloudtrail-log-sf	ObjectCount	12264
V_1	sample-cmh-exclude	11/3/2020	546264889236	us-west-1	STANDARD	PREFIX	AWSLogs%2F546264889236%2FCloudTrail%2Fus	cloudtrail-log-sf	ReplicatedStorageBytes	0
V_1	sample-cmh-exclude	11/3/2020	546264889236	us-west-1	STANDARD	PREFIX	AWSLogs%2F546264889236%2FCloudTrail%2Fus	cloudtrail-log-sf	ReplicatedObjectCount	0
V_1	sample-cmh-exclude	11/3/2020	546264889236	us-west-1	STANDARD	PREFIX	AWSLogs%2F546264889236%2FCloudTrail%2Fus	cloudtrail-log-sf	EncryptedStorageBytes	29996800
V_1	sample-cmh-exclude	11/3/2020	546264889236	us-west-1	STANDARD	PREFIX	AWSLogs%2F546264889236%2FCloudTrail%2Fus	cloudtrail-log-sf	EncryptedObjectCount	12264
V_1	sample-cmh-exclude	11/3/2020	546264889236	us-west-1	STANDARD	PREFIX	AWSLogs%2F546264889236%2FCloudTrail%2Fus	cloudtrail-log-sf	DeleteMarkerObjectCount	0
V_1	sample-cmh-exclude	11/3/2020	546264889236	us-west-1	STANDARD	PREFIX	AWSLogs%2F546264889236%2FCloudTrail%2Fus	cloudtrail-log-sf	ObjectLockEnabledStorageBytes	0
V_1	sample-cmh-exclude	11/3/2020	546264889236	us-west-1	STANDARD	PREFIX	AWSLogs%2F546264889236%2FCloudTrail%2Fus	cloudtrail-log-sf	ObjectLockEnabledObjectCount	0
V_1	sample-cmh-exclude	11/3/2020	546264889236	us-west-1	STANDARD	PREFIX	AWSLogs%2F546264889236%2FCloudTrail%2Fus	cloudtrail-log-sf	CurrentVersionStorageBytes	29996800
V_1	sample-cmh-exclude	11/3/2020	546264889236	us-west-1	STANDARD	PREFIX	AWSLogs%2F546264889236%2FCloudTrail%2Fus	cloudtrail-log-sf	CurrentVersionObjectCount	12264
V_1	sample-cmh-exclude	11/3/2020	546264889236	us-west-1	STANDARD	PREFIX	AWSLogs%2F546264889236%2FCloudTrail%2Fus	cloudtrail-log-sf	NonCurrentVersionStorageBytes	0
V_1	sample-cmh-exclude	11/3/2020	546264889236	us-west-1	STANDARD	PREFIX	AWSLogs%2F546264889236%2FCloudTrail%2Fus	cloudtrail-log-sf	NonCurrentVersionObjectCount	0
V_1	sample-cmh-exclude	11/3/2020	546264889236	us-west-1	STANDARD	PREFIX	AWSLogs%2F546264889236%2FCloudTrail%2Fus	cloudtrail-log-sf	IncompleteMultipartUploadStorageBytes	0
V_1	sample-cmh-exclude	11/3/2020	546264889236	us-west-1	STANDARD	PREFIX	AWSLogs%2F546264889236%2FCloudTrail%2Fus	cloudtrail-log-sf	IncompleteMultipartUploadObjectCount	0

A continuación se muestra un ejemplo de una exportación de métricas de Lente de almacenamiento de S3 con datos de grupos de Lente de almacenamiento.

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

Monitoreo de métricas de S3 Storage Lens en CloudWatch

Puede publicar métricas de S3 Storage Lens en Amazon CloudWatch para crear una vista unificada del estado operativo en los paneles de CloudWatch. También puede utilizar las características de CloudWatch, como alarmas y acciones desencadenadas, cálculos de métricas y detección de anomalías para monitorear y tomar medidas en las métricas de S3 Storage Lens. Además, las operaciones de la API de CloudWatch habilitan las aplicaciones, incluidos los proveedores de

terceros, para acceder a las métricas de S3 Storage Lens. Para obtener más información acerca de las características de CloudWatch, consulte la Guía del usuario de Amazon CloudWatch.

Puede habilitar la opción de publicación de CloudWatch para configuraciones de panel nuevas o existentes mediante la consola de Amazon S3, la API de REST de Amazon S3, AWS CLI y los SDK de AWS. Los paneles actualizados a métricas y recomendaciones avanzadas de S3 Storage Lens pueden utilizar la opción de publicación de CloudWatch. Para conocer los precios de métricas y recomendaciones avanzadas de S3 Storage Lens, consulte Precios de Amazon S3. No se aplican cargos por publicación de métricas de CloudWatch adicionales; sin embargo, se aplican otros cargos de CloudWatch, como paneles, alarmas y llamadas a la API. Para obtener más información, consulte los precios de Amazon CloudWatch.

Las métricas de S3 Storage Lens se publican en CloudWatch, en la cuenta que es propietaria de la configuración de S3 Storage Lens. Después de habilitar la opción de publicación de CloudWatch dentro de métricas y recomendaciones avanzadas, puede acceder a las métricas de nivel de organización, cuenta y bucket en CloudWatch. Las métricas de nivel de prefijo no están disponibles en CloudWatch.

> **Note**
>
> Las métricas de S3 Storage Lens son métricas diarias y se publican en CloudWatch una vez al día. Cuando consulte métricas de S3 Storage Lens en CloudWatch, el periodo de la consulta debe ser de 1 día (86 400 segundos). Después de que las métricas diarias de S3 Storage Lens aparezcan en el panel de S3 Storage Lens de la consola de Amazon S3, pueden pasar unas horas para que aparezcan esas mismas métricas en CloudWatch. Cuando habilita la opción de publicación de CloudWatch para las métricas de S3 Storage Lens por primera vez, pueden pasar hasta 24 horas para que las métricas se publiquen en CloudWatch.

Después de habilitar la opción de publicación de CloudWatch, puede utilizar las siguientes características de CloudWatch para monitorear y analizar los datos de S3 Storage Lens:

- Paneles: utilice paneles de CloudWatch para crear paneles personalizados de S3 Storage Lens. Comparta el panel de CloudWatch con personas que no tienen acceso directo a la Cuenta de AWS, con los equipos, con partes interesadas y personas ajenas a la organización.
- Alarmas y acciones desencadenadas: configure alarmas que vigilen las métricas y tomen medidas cuando se traspase un umbral. Por ejemplo, puede configurar una alarma que envíe una

notificación de Amazon SNS cuando la métrica de Incomplete Multipart Upload Bytes (Bytes de carga multiparte incompletos) supere 1 GB durante tres días consecutivos.

- Detección de anomalías: habilite la detección de anomalías para analizar de manera continua las métricas, determinar los valores de referencia normales y descubrir anomalías. Puede crear una alarma de detección de anomalías basada en el valor esperado de una métrica. Por ejemplo, puede monitorear las anomalías de la métrica de Object Lock Enabled Bytes (Bytes habilitados para Object Lock) con el objetivo de detectar la eliminación no autorizada de la configuración de Object Lock.
- Cálculo de métricas: también puede utilizar cálculo de métricas para consultar varias métricas de S3 Storage Lens y utilizar expresiones matemáticas para crear nuevas series temporales basadas en estas métricas. Por ejemplo, puede crear una nueva métrica para obtener el tamaño de objeto promedio dividiendo `StorageBytes` por `ObjectCount`.

Para obtener más información acerca de la opción de publicación de CloudWatch para las métricas de S3 Storage Lens, consulte los siguientes temas:

Temas

- Métricas y dimensiones de S3 Storage Lens
- Habilitación de publicación de CloudWatch para S3 Storage Lens
- Trabajo con métricas de S3 Storage Lens en CloudWatch

Métricas y dimensiones de S3 Storage Lens

Para enviar métricas de S3 Storage Lens a CloudWatch, debe habilitar la opción de publicación de CloudWatch en las métricas y recomendaciones avanzadas de S3 Storage Lens. Cuando las métricas avanzadas estén habilitadas, puede utilizar los paneles de CloudWatch para monitorear las métricas de S3 Storage Lens junto con otras métricas de aplicaciones y crear una vista unificada del estado operativo. Puede utilizar dimensiones para filtrar las métricas de S3 Storage Lens en CloudWatch por organización, cuenta, bucket, clase de almacenamiento, región e ID de configuración de métricas.

Las métricas de S3 Storage Lens se publican en CloudWatch, en la cuenta que es propietaria de la configuración de S3 Storage Lens. Después de habilitar la opción de publicación de CloudWatch dentro de métricas y recomendaciones avanzadas, puede acceder a las métricas de nivel de organización, cuenta y bucket en CloudWatch. Las métricas de nivel de prefijo no están disponibles en CloudWatch.

> **Note**
>
> Las métricas de S3 Storage Lens son métricas diarias y se publican en CloudWatch una vez al día. Cuando consulte métricas de S3 Storage Lens en CloudWatch, el periodo de la consulta debe ser de 1 día (86 400 segundos). Después de que las métricas diarias de S3 Storage Lens aparezcan en el panel de S3 Storage Lens de la consola de Amazon S3, pueden pasar unas horas para que aparezcan esas mismas métricas en CloudWatch. Cuando habilita la opción de publicación de CloudWatch para las métricas de S3 Storage Lens por primera vez, pueden pasar hasta 24 horas para que las métricas se publiquen en CloudWatch.

Para obtener más información acerca de las métricas y dimensiones de S3 Storage Lens en CloudWatch, consulte los siguientes temas.

Temas

- Métricas
- Dimensiones

Métricas

Las métricas de S3 Storage Lens están disponibles como métricas en CloudWatch. Las métricas de S3 Storage Lens se publican en el espacio de nombres de `AWS/S3/Storage-Lens`. Este espacio de nombres es solo para métricas de S3 Storage Lens. Las métricas de bucket, solicitud y replicación de Amazon S3 se publican en el espacio de nombres de `AWS/S3`.

Las métricas de S3 Storage Lens se publican en CloudWatch, en la cuenta que es propietaria de la configuración de S3 Storage Lens. Después de habilitar la opción de publicación de CloudWatch dentro de métricas y recomendaciones avanzadas, puede acceder a las métricas de nivel de organización, cuenta y bucket en CloudWatch. Las métricas de nivel de prefijo no están disponibles en CloudWatch.

En S3 Storage Lens, las métricas se agregan y almacenan solo en la región de origen designada. Las métricas de S3 Storage Lens también se publican en CloudWatch en la región de origen que especifica en la configuración de S3 Storage Lens.

Para obtener una lista completa de las métricas de S3 Storage Lens, incluida una lista de las métricas disponibles en CloudWatch, consulte Glosario de métricas de Amazon S3 Storage Lens.

> **Note**
>
> La estadística válida para las métricas de S3 Storage Lens en CloudWatch es Average (Promedio). Para obtener más información acerca de las estadísticas en CloudWatch, consulte Definiciones de estadísticas de CloudWatch en la Guía del usuario de Amazon CloudWatch.

Grado de detalle de las métricas de S3 Storage Lens en CloudWatch

S3 Storage Lens ofrece métricas pormenorizadas de organización, cuenta, bucket y prefijo. S3 Storage Lens publica métricas de S3 Storage Lens de organización, cuenta y bucket en CloudWatch. Las métricas de S3 Storage Lens de nivel de prefijo no están disponibles en CloudWatch.

Para obtener más información acerca del grado de detalle de las métricas de S3 Storage Lens disponibles en CloudWatch, consulte la siguiente lista:

- Organización: métricas agregadas en las cuentas de miembros de la organización. S3 Storage Lens publica métricas de cuentas de miembros en CloudWatch en la cuenta de administración.
  - Organización y cuenta: métricas de las cuentas de miembros de la organización.
  - Organización y bucket: métricas de buckets de Amazon S3 en las cuentas de miembros de la organización.
- Cuenta (nivel no organizativo): métricas agregadas en los buckets de la cuenta.
- Bucket (nivel no organizativo): métricas de un bucket específico. En CloudWatch, S3 Storage Lens publica estas métricas en la Cuenta de AWS que creó la configuración de S3 Storage Lens. S3 Storage Lens publica estas métricas solo para configuraciones que no sean organizaciones.

Dimensiones

Cuando S3 Storage Lens envía datos a CloudWatch, se adjuntan las dimensiones a cada métrica. Las dimensiones son categorías que describen las características de las métricas. Puede utilizar dimensiones para filtrar los resultados que muestra CloudWatch.

Por ejemplo, todas las métricas de S3 Storage Lens de CloudWatch tienen la dimensión `configuration_id`. Puede utilizar esta dimensión para distinguir entre métricas asociadas a una configuración específica de S3 Storage Lens. El `organization_id` identifica métricas de organización. Para obtener más información acerca de las dimensiones en CloudWatch, consulte Dimensiones en la Guía del usuario de CloudWatch.

Existen diferentes dimensiones disponibles para las métricas de S3 Storage Lens en función del grado de detalle de las métricas. Por ejemplo, puede utilizar la dimensión `organization_id` para filtrar las métricas de organización por ID de AWS Organizations. Sin embargo, no puede utilizar esta dimensión para las métricas de bucket y cuenta. Para obtener más información, consulte Filtrado de métricas mediante dimensiones.

Para ver qué dimensiones están disponibles para la configuración de S3 Storage Lens, consulte la siguiente tabla:

Dimensión	Descripción	Bucket	Cuenta	Organización	Organización y bucket	Organización y cuenta
`configuration_id`	El nombre del panel de la configuración de S3 Storage Lens que se indica en las métricas	.	.	.	.	.
`metrics_version`	La versión de las métricas de S3 Storage Lens. La versión de las métricas tiene un valor fijo de `1.0`.	.	.	.	.	.
`organization_id`	El ID de AWS Organizations para las métricas	.	.	.	.	.
`aws_account_number`	La Cuenta de AWS asociada a las métricas	.	.	.	.	.
`aws_region`	La Región de AWS para las métricas	.	.	.	.	.
`bucket_name`	El nombre del bucket de S3 que se indica en las métricas	.	.	.	.	.
`storage_class`	La clase de almacenamiento del bucket indicado en las métricas	.	.	.	.	.
`record_type`	El grado de detalle de las métricas: ORGANIZACIÓN, CUENTA, BUCKET	BUCKET	CUENTA	ORGANIZACIÓN	.	.

Habilitación de publicación de CloudWatch para S3 Storage Lens

Puede publicar métricas de S3 Storage Lens en Amazon CloudWatch para crear una vista unificada del estado operativo en los paneles de CloudWatch. También puede utilizar las características de CloudWatch, como alarmas y acciones desencadenadas, cálculos de métricas y detección de anomalías para monitorear y tomar medidas en las métricas de S3 Storage Lens. Además, las operaciones de la API de CloudWatch habilitan las aplicaciones, incluidos los proveedores de terceros, para acceder a las métricas de S3 Storage Lens. Para obtener más información acerca de las características de CloudWatch, consulte la Guía del usuario de Amazon CloudWatch.

Las métricas de S3 Storage Lens se publican en CloudWatch, en la cuenta que es propietaria de la configuración de S3 Storage Lens. Después de habilitar la opción de publicación de CloudWatch dentro de métricas y recomendaciones avanzadas, puede acceder a las métricas de nivel de organización, cuenta y bucket en CloudWatch. Las métricas de nivel de prefijo no están disponibles en CloudWatch.

Puede habilitar el soporte de CloudWatch para configuraciones de panel nuevas o existentes mediante la consola de S3, las API de REST de Amazon S3, AWS CLI y SDK de AWS. La opción de publicación de CloudWatch está disponible para los paneles que se actualizan a métricas y recomendaciones avanzadas de S3 Storage Lens. Para conocer los precios de métricas y recomendaciones avanzadas de S3 Storage Lens, consulte Precios de Amazon S3. No se aplican cargos por publicación de métricas de CloudWatch adicionales; sin embargo, se aplican otros cargos de CloudWatch, como paneles, alarmas y llamadas a la API.

Para habilitar la opción de publicación de CloudWatch para las métricas de S3 Storage Lens, consulte los siguientes temas.

> **Note**
>
> Las métricas de S3 Storage Lens son métricas diarias y se publican en CloudWatch una vez al día. Cuando consulte métricas de S3 Storage Lens en CloudWatch, el periodo de la consulta debe ser de 1 día (86 400 segundos). Después de que las métricas diarias de S3 Storage Lens aparezcan en el panel de S3 Storage Lens de la consola de Amazon S3, pueden pasar unas horas para que aparezcan esas mismas métricas en CloudWatch. Cuando habilita la opción de publicación de CloudWatch para las métricas de S3 Storage Lens por primera vez, pueden pasar hasta 24 horas para que las métricas se publiquen en CloudWatch.

Actualmente, las métricas de S3 Storage Lens no se pueden consumir a través de las transmisiones de CloudWatch.

Uso de la consola de S3

Cuando actualiza un panel de S3 Storage Lens, no puede cambiar el nombre ni la región de origen del panel. Tampoco se puede cambiar el alcance del panel predeterminado, que tiene como alcance el almacenamiento completo de la cuenta.

Para actualizar un panel de S3 Storage Lens a fin de habilitar la publicación de CloudWatch

1. Inicie sesión en la AWS Management Console y abra la consola de Amazon S3 en https://console.aws.amazon.com/s3/.
2. En el panel de navegación, elija Storage Lens, Dashboards (Paneles).
3. Elija el panel que desea editar y, a continuación, elija Edit (Editar).
4. En Metrics selection (Selección de métricas), elija Advanced metrics and recommendations (Métricas y recomendaciones avanzadas).

   Las métricas y recomendaciones avanzadas se encuentran disponibles por un cargo adicional. Esta opción incluye un periodo de 15 meses para consultas de datos, métricas de uso agregadas a nivel de prefijo y métricas de actividad agregadas por bucket, la opción de publicación de CloudWatch y recomendaciones contextuales que le ayudan a optimizar los costos de almacenamiento y aplicar las prácticas recomendadas de protección de datos. Para obtener más información, consulte Precios de Amazon S3.
5. En Select Advanced metrics and recommendations features (Seleccionar características de métricas y recomendaciones avanzadas), seleccione CloudWatch publishing (Publicación de CloudWatch).

   Important

   Si la configuración habilita la agregación de prefijos para métricas de uso, las métricas de prefijo no se publicarán en CloudWatch. En CloudWatch, solo se publican métricas de S3 Storage Lens a nivel de bucket, cuenta y organización.
6. Elija Save changes (Guardar cambios).

Para crear un nuevo panel de S3 Storage Lens que habilite la compatibilidad con CloudWatch

1. Inicie sesión en la AWS Management Console y abra la consola de Amazon S3 en https://console.aws.amazon.com/s3/.
2. En el panel de navegación izquierdo, elija Storage Lens, Dashboards (Paneles).
3. Elija Create dashboard (Crear un panel).
4. En General, defina las opciones de configuración siguientes:

   a. Para Dashboard name (Nombre del panel), ingrese el nombre del panel.

      Los nombres de los paneles deben tener menos de 65 caracteres y no deben contener caracteres ni espacios especiales. No puede cambiar el nombre del panel una vez que lo creó.

   b. Elija la región de origen del panel.

      Las métricas para todas las regiones incluidas en este alcance del panel se almacenan de forma centralizada en la región de origen designada. En CloudWatch, las métricas de S3 Storage Lens también están disponibles en la región de origen. No puede cambiar la región principal una vez que se creó el panel.
5. (Opcional) Para agregar etiquetas, elija Add tag (Agregar etiqueta) e ingrese la Key (Clave) y el Value (Valor) de la etiqueta.

   > **Note**
   >
   > Puede agregar hasta 50 etiquetas a la configuración del panel.
6. Defina el alcance de la configuración:

   a. Si va a crear una configuración de organización, elija las cuentas que desea incluir en la configuración: Include all accounts in your configuration (Incluir todas las cuentas de la configuración) o Limit the scope to your signed-in account (Limitar el alcance a la cuenta que ha iniciado sesión).

      > **Note**
      >
      > Cuando crea una configuración de organización que incluye todas las cuentas, solo puede incluir o excluir regiones y no buckets.

b. Elija las regiones y los buckets que desea que S3 Storage Lens incluya en la configuración del panel realizando lo siguiente:

- Para incluir todas las regiones, elija Include Regions and buckets (Incluir regiones y buckets).
- Para incluir regiones específicas, borre Include all Regions (Incluir todas las regiones). En Choose Regions to include (Elegir las regiones que desea incluir), elija las regiones que desea que S3 Storage Lens incluya en el panel.
- Para incluir buckets específicos, borre Include all buckets (Incluir todos los buckets). En Choose buckets to include (Elegir los buckets que desea incluir), elija los buckets que desea que S3 Storage Lens incluya en el panel.

> **Note**
>
> Puede elegir hasta 50 buckets.

7. Para Metrics selection (Selección de métricas), elija Advanced metrics and recommendations (Métricas y recomendaciones avanzadas).

   Para obtener más información sobre los precios de métricas y recomendaciones avanzadas, consulte Precios de Amazon S3.

8. En Advanced metrics and recommendations features (Características de métricas y recomendaciones avanzadas), seleccione las opciones que desea habilitar:

   - Advanced metrics (Métricas avanzadas)
   - Publicación de CloudWatch

   > **Important**
   >
   > Si habilita la agregación de prefijos para la configuración de S3 Storage Lens, las métricas de nivel de prefijo no se publicarán en CloudWatch. En CloudWatch, solo se publican métricas de S3 Storage Lens a nivel de bucket, cuenta y organización.

   - Agregación de prefijos

> **Note**
>
> Para obtener más información sobre características de métricas y recomendaciones avanzadas, consulte Selección de métricas.

9. Si ha habilitado Advanced metrics (Métricas avanzadas), seleccione las Advanced metrics categories (Categorías de métricas avanzadas) que desea mostrar en el panel de S3 Storage Lens:

   - Métricas de actividad
   - Detailed status code metrics (Métricas de código de estado detalladas)
   - Advanced cost optimization metrics (Métricas de optimización de costos avanzadas)
   - Advanced data protection metrics (Métricas de protección de datos avanzadas)

   Para obtener más información sobre categorías de métricas, consulte Categorías de métricas. Para obtener una lista completa de métricas, consulte Glosario de métricas de Amazon S3 Storage Lens.

10. (Opcional) Configure la exportación de métricas.

    Para obtener más información sobre cómo configurar la exportación de métricas, consulte el paso Creación de un panel de Amazon S3 Storage Lens.

11. Elija Create dashboard (Crear un panel).

Utilización de la AWS CLI

El siguiente ejemplo de AWS CLI habilita la opción de publicación de CloudWatch mediante una configuración de métricas y recomendaciones avanzadas de S3 Storage Lens de organización. Para utilizar este ejemplo, reemplace los *`user input placeholders`* con su propia información.

```
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.
    }
  }
}
```

Uso de AWS SDK para 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();
        }
    }
}
```

Uso de la API de REST

Para habilitar la opción de publicación de CloudWatch mediante la API de REST de Amazon S3, puede utilizar PutStorageLensConfiguration.

Pasos siguientes

Después de habilitar la opción de publicación de CloudWatch, puede acceder a las métricas de S3 Storage Lens en CloudWatch. También puede aprovechar las características de CloudWatch para monitorear y analizar los datos de S3 Storage Lens en CloudWatch. Para obtener más información, consulte los siguientes temas:

- Métricas y dimensiones de S3 Storage Lens
- Trabajo con métricas de S3 Storage Lens en CloudWatch

Trabajo con métricas de S3 Storage Lens en CloudWatch

Puede publicar métricas de S3 Storage Lens en Amazon CloudWatch para crear una vista unificada del estado operativo en los paneles de CloudWatch. También puede utilizar las características de CloudWatch, como alarmas y acciones desencadenadas, cálculos de métricas y detección de anomalías para monitorear y tomar medidas en las métricas de S3 Storage Lens. Además, las operaciones de la API de CloudWatch habilitan las aplicaciones, incluidos los proveedores de terceros, para acceder a las métricas de S3 Storage Lens. Para obtener más información acerca de las características de CloudWatch, consulte la Guía del usuario de Amazon CloudWatch.

Puede habilitar la opción de publicación de CloudWatch para configuraciones de panel nuevas o existentes mediante la consola de Amazon S3, las API de REST de Amazon S3, AWS CLI y SDK de AWS. La opción de publicación de CloudWatch está disponible para los paneles que se actualizan a métricas y recomendaciones avanzadas de S3 Storage Lens. Para conocer los precios de métricas y recomendaciones avanzadas de S3 Storage Lens, consulte Precios de Amazon S3. No se aplican cargos por publicación de métricas de CloudWatch adicionales; sin embargo, se aplican otros cargos de CloudWatch, como paneles, alarmas y llamadas a la API. Para obtener más información, consulte los precios de Amazon CloudWatch.

Las métricas de S3 Storage Lens se publican en CloudWatch, en la cuenta que es propietaria de la configuración de S3 Storage Lens. Después de habilitar la opción de publicación de CloudWatch dentro de métricas y recomendaciones avanzadas, puede acceder a las métricas de nivel de organización, cuenta y bucket en CloudWatch. Las métricas de nivel de prefijo no están disponibles en CloudWatch.

> **Note**
>
> Las métricas de S3 Storage Lens son métricas diarias y se publican en CloudWatch una vez al día. Cuando consulte métricas de S3 Storage Lens en CloudWatch, el periodo de la consulta debe ser de 1 día (86 400 segundos). Después de que las métricas diarias de S3 Storage Lens aparezcan en el panel de S3 Storage Lens de la consola de Amazon S3, pueden pasar unas horas para que aparezcan esas mismas métricas en CloudWatch. Cuando habilita la opción de publicación de CloudWatch para las métricas de S3 Storage Lens por primera vez, pueden pasar hasta 24 horas para que las métricas se publiquen en CloudWatch.

Actualmente, las métricas de S3 Storage Lens no se pueden consumir a través de las transmisiones de CloudWatch.

Para obtener más información sobre cómo trabajar con métricas de S3 Storage Lens en CloudWatch, consulte los siguientes temas.

Temas

- Trabajo con paneles de CloudWatch
- Configuración de alarmas, activación de acciones y uso de la detección de anomalías
- Filtrado de métricas mediante dimensiones
- Cómputo de nuevas métricas con cálculo de métricas
- Uso de expresiones de búsqueda en gráficos

Trabajo con paneles de CloudWatch

Puede utilizar paneles de CloudWatch para monitorear las métricas de S3 Storage Lens junto con otras métricas de aplicaciones y crear una vista unificada del estado operativo. Los paneles son páginas de inicio personalizables en la consola de CloudWatch que puede utilizar para monitorear sus recursos en una única vista.

CloudWatch tiene un amplio control de permisos que no admite la restricción del acceso a un conjunto específico de métricas o dimensiones. Los usuarios de la cuenta u organización que tengan acceso a CloudWatch podrán ver las métricas de todas las configuraciones de S3 Storage Lens en las que está habilitada la opción de soporte de CloudWatch. No es posible administrar los permisos para paneles específicos como se hace en S3 Storage Lens. Para obtener más información acerca de los permisos de CloudWatch, consulte Managing access permissions to your CloudWatch resources (Administración de permisos de acceso para los recursos de CloudWatch) en la Guía del usuario de Amazon CloudWatch.

Para obtener más información acerca del uso de paneles de CloudWatch y la configuración de permisos, consulte Uso de paneles de Amazon CloudWatch y Compartir paneles de CloudWatch en la Guía del usuario de Amazon CloudWatch.

Configuración de alarmas, activación de acciones y uso de la detección de anomalías

Puede configurar alarmas de CloudWatch que vigilen las métricas de S3 Storage Lens en CloudWatch y adopten medidas cuando se traspase un umbral. Por ejemplo, puede configurar una

alarma que envíe una notificación de Amazon SNS cuando la métrica de Incomplete Multipart Upload Bytes (Bytes de carga multiparte incompletos) supere 1 GB durante tres días consecutivos.

También puede habilitar la detección de anomalías para analizar de manera continua las métricas de S3 Storage Lens, determinar los valores de referencia normales y descubrir anomalías. Puede crear una alarma de detección de anomalías en función del valor esperado de una métrica. Por ejemplo, puede monitorear las anomalías de la métrica de Object Lock Enabled Bytes (Bytes habilitados para Object Lock) con el objetivo de detectar la eliminación no autorizada de la configuración de Object Lock.

Para obtener más información y ejemplos, consulte Uso de las alarmas de Amazon CloudWatch y Creación de una alarma desde una métrica en un gráfico en la Guía del usuario de Amazon CloudWatch.

Filtrado de métricas mediante dimensiones

Puede utilizar dimensiones para filtrar las métricas de S3 Storage Lens en la consola de CloudWatch. Por ejemplo, puede filtrar por `configuration_id`, `aws_account_number`, `aws_region`, `bucket_name` y otros.

S3 Storage Lens admite varias configuraciones de panel por cada cuenta. Esto significa que las distintas configuraciones pueden incluir el mismo bucket. Cuando estas métricas se publiquen en CloudWatch, el bucket tendrá métricas duplicadas dentro de CloudWatch. Puede utilizar la dimensión `configuration_id` para ver métricas solo de una configuración específica de S3 Storage Lens en CloudWatch. Cuando filtra por `configuration_id`, solo ve las métricas que están asociadas a la configuración que identifica.

Para obtener más información acerca del filtrado por ID de configuración, consulte Búsqueda de métricas disponibles en la Guía del usuario de Amazon CloudWatch.

Cómputo de nuevas métricas con cálculo de métricas

Es posible utilizar cálculo de métricas para consultar varias métricas de S3 Storage Lens y utilizar expresiones matemáticas para crear nuevas series temporales basadas en estas métricas. Por ejemplo, se puede crear una nueva métrica para objetos sin cifrar restando objetos cifrados del recuento de objetos. También es posible crear una métrica para obtener el tamaño de objeto promedio dividiendo `StorageBytes` por `ObjectCount` o el número de bytes a los que se accede un día dividiendo `BytesDownloaded` por `StorageBytes`.

Para obtener más información, consulte Uso de cálculo de métricas en la Guía del usuario de Amazon CloudWatch.

Uso de expresiones de búsqueda en gráficos

Con las métricas de S3 Storage Lens, puede crear una expresión de búsqueda. Por ejemplo, puede crear una expresión de búsqueda para todas las métricas denominada IncompleteMultipartUploadStorageBytes y agregar SUM a la expresión. Con esta expresión de búsqueda, puede ver el total de bytes cargados multiparte incompletos en todas las dimensiones del almacenamiento en una sola métrica.

En este ejemplo se muestra la sintaxis que utilizaría para crear una expresión de búsqueda para todas las métricas denominada IncompleteMultipartUploadStorageBytes.

```
SUM(SEARCH('{AWS/S3/Storage-
Lens,aws_account_number,aws_region,configuration_id,metrics_version,record_type,storage_class}
 MetricName="IncompleteMultipartUploadStorageBytes"', 'Average',86400))
```

Para obtener más información acerca de esta sintaxis, consulte Sintaxis de expresiones de búsqueda de CloudWatch en la Guía del usuario de Amazon CloudWatch. Para crear un gráfico de CloudWatch con una expresión de búsqueda, consulte Creación de un gráfico de CloudWatch con una expresión de búsqueda en la Guía del usuario de Amazon CloudWatch.

Casos de uso de métricas de lente de almacenamiento de Amazon S3

Puede usar el panel de lente de almacenamiento de Amazon S3 para visualizar la información y las tendencias, marcar los valores atípicos y recibir recomendaciones. Las métricas de S3 Storage Lens se organizan en categorías que se alinean con los casos de uso claves. Puede utilizar estas métricas para hacer lo siguiente:

- Identificar oportunidades de optimización de costos
- Aplicar prácticas recomendadas de protección de datos
- Aplicar prácticas recomendadas de administración de acceso
- Mejorar el rendimiento de las cargas de trabajo de las aplicaciones

Por ejemplo, con las métricas de optimización de costos, puede identificar oportunidades para reducir los costos de almacenamiento de Amazon S3. Puede identificar buckets con cargas multipartes con más de 7 días de antigüedad o buckets que acumulan versiones no actuales.

Del mismo modo, puede utilizar las métricas de protección de datos para identificar los buckets que no siguen las prácticas recomendadas de protección de datos de la organización. Por ejemplo,

puede identificar los buckets que no usan claves de AWS Key Management Service (SSE-KMS) para el cifrado predeterminado o que no tienen habilitado el control de versiones de S3.

Con las métricas de administración de acceso de S3 Storage Lens, puede identificar la configuración de los bucket para la propiedad de objetos de S3 para poder migrar los permisos de la lista de control de acceso (ACL) a las políticas de bucket y desactivar las ACL.

Si tiene la opción S3 Storage Lens advanced metrics (Métricas avanzadas de S3 Storage Lens) habilitada, puede usar las métricas de código de estado detalladas para obtener recuentos para solicitudes exitosas o erróneas que puede usar para solucionar problemas de acceso o rendimiento.

Con las métricas avanzadas, también puede acceder a métricas adicionales de optimización de costos y protección de datos que puede utilizar para identificar oportunidades a fin de reducir aún más los costos generales de almacenamiento de S3 y alinearse mejor con las prácticas recomendadas para proteger los datos. Por ejemplo, las métricas de optimización de costos avanzadas incluyen recuentos de reglas del ciclo de vida que puede usar para identificar buckets que no tienen reglas del ciclo de vida para que hagan vencer las cargas multipartes incompletas que tengan más de 7 días de antigüedad. Las métricas de protección de datos avanzadas incluyen recuentos de reglas de replicación.

Para obtener más información sobre categorías de métricas, consulte Categorías de métricas. Para obtener una lista completa de las métricas de S3 Storage Lens, consulte Glosario de métricas de Amazon S3 Storage Lens.

Temas

- Uso de Amazon S3 Storage Lens para optimizar sus costos de almacenamiento
- Uso de S3 Storage Lens para proteger sus datos
- Uso de S3 Storage Lens para auditar la configuración de la propiedad de objetos
- Uso de métricas de S3 Storage Lens para mejorar el rendimiento

Uso de Amazon S3 Storage Lens para optimizar sus costos de almacenamiento

Puede utilizar las métricas de optimización de costos de Lente de almacenamiento de S3 para reducir el costo total del almacenamiento de S3. Las métricas de optimización de costos pueden ayudarle a confirmar que ha configurado Amazon S3 de forma rentable y de acuerdo con las prácticas recomendadas. Por ejemplo, puede identificar las siguientes oportunidades de optimización de costos:

- Buckets con cargas multiparte incompletas de más de 7 días
- Buckets que acumulan numerosas versiones no actuales
- Buckets que no tienen reglas de ciclo de vida para anular cargas multiparte incompletas
- Buckets que no tienen reglas de ciclo de vida para hacer vencer objetos de versiones no actuales
- Buckets que no tienen reglas de ciclo de vida para transferir objetos a una clase de almacenamiento diferente

A continuación, puede utilizar estos datos para agregar reglas de ciclo de vida adicionales a los buckets.

Los siguientes ejemplos muestran cómo puede utilizar métricas de optimización de costos en el panel de Lente de almacenamiento de S3 para optimizar los costos de almacenamiento.

Temas

Identifique sus buckets más grandes de S3

Usted paga por almacenar objetos en los buckets de S3. La tarifa que se le cobra depende del tamaño de los objetos, el tiempo que se almacenan y las clases de almacenamiento. Con Lente de almacenamiento de S3, obtendrá una vista centralizada de todos los buckets de la cuenta. Para ver todos los buckets de todas las cuentas de su organización, configure un panel de S3 Storage Lens de AWS Organizations. Desde la vista del panel, puede identificar los buckets más grandes.

Paso 1: identificar los buckets más grandes

1. Inicie sesión en la AWS Management Console y abra la consola de Amazon S3 en https://console.aws.amazon.com/s3/.
2. En el panel de navegación izquierdo, elija Storage Lens, Dashboards (Paneles).
3. En la lista Dashboards (Paneles), elija el panel que desea ver.

Cuando se abre el panel, puede ver la fecha más reciente para la que Lente de almacenamiento de S3 ha recopilado métricas. El panel siempre se carga en la fecha más reciente para la que están disponibles las métricas.

4. Para ver una clasificación de los buckets más grandes de la métrica de Total storage (Almacenamiento total) para un intervalo de fechas seleccionado, desplácese hacia abajo hasta la sección Top N overview for date (Información general de N principales para fecha).

   Puede cambiar el orden de clasificación para mostrar los buckets más pequeños. También puede ajustar la selección de Metric (Métricas) para clasificar los grupos según cualquiera de las métricas disponibles. La sección Top N overview for date (Información general de las N principales para fechas) también muestra el cambio porcentual con respecto al día o la semana anteriores y un minigráfico para visualizar la tendencia. Esta tendencia es una tendencia de 14 días para las métricas gratuitas y una tendencia de 30 días para las métricas y recomendaciones avanzadas.

   > **Note**
   >
   > Con las métricas y recomendaciones avanzadas de Lente de almacenamiento de S3, las métricas están disponibles para consultas durante 15 meses. Para obtener más información, consulte Selección de métricas.

5. Para obtener información más detallada sobre los buckets, desplácese hasta la parte superior de la página y, a continuación, elija la pestaña Bucket.

   En la pestaña Bucket, puede ver detalles, como la tasa de crecimiento reciente, el tamaño promedio del objeto, los prefijos más grandes y la cantidad de objetos.

Paso 2: navegar hasta los buckets e investigar

Después de haber identificado los buckets de S3 más grandes, puede navegar a cada uno de ellos dentro de la consola de S3 para ver los objetos en el bucket, entender su carga de trabajo asociada e identificar a sus propietarios internos. Puede contactar con los propietarios del bucket para descubrir si se espera el crecimiento o si el crecimiento necesita más monitoreo y control.

Descubra buckets en frío de Amazon S3

Si tiene la opción de métricas avanzadas de S3 Storage Lens habilitada, puede usar las métricas de actividad para comprender lo fríos que están los buckets de S3. Un bucket “inactivo” es aquel a cuyo

almacenamiento ya no se accede (o se accede con muy poca frecuencia). Esta falta de actividad suele indicar que no se accede con frecuencia a los objetos del bucket.

Las métricas de actividad, como GET Requests (Solicitudes GET) y Download Bytes (Bytes de descarga), indican la frecuencia con la que se accede a los buckets cada día. Para comprender la consistencia del patrón de acceso y detectar buckets a los que ya no se accede, actualice estos datos durante varios meses. La métrica de frecuencia de recuperación, que se calcula como bytes de descarga/almacenamiento total, indica la proporción de almacenamiento en un bucket al que se accede diariamente.

> **Note**
>
> Los bytes de descarga se duplican en los casos en que el mismo objeto se descarga varias veces durante el día.

Requisito previo

Para ver métricas de actividad en el panel de Lente de almacenamiento de S3, debe habilitar Advanced metrics and recommendations (Métricas y recomendaciones avanzadas) de Lente de almacenamiento de S3 y, a continuación, seleccionar Activity metrics (Métricas de actividad). Para obtener más información, consulte Creación y actualización de los paneles de Amazon S3 Storage Lens.

Paso 1: identificar buckets activos

1. Inicie sesión en la AWS Management Console y abra la consola de Amazon S3 en https://console.aws.amazon.com/s3/.
2. En el panel de navegación izquierdo, elija Storage Lens, Dashboards (Paneles).
3. En la lista Dashboards (Paneles), elija el panel que desea ver.
4. Elija la pestaña Bucket y, a continuación, desplácese hacia abajo hasta la sección Bubble analysis by buckets for date (Análisis de burbujas por buckets para fechas).

   En la sección Bubble analysis by buckets for date (Análisis de burbujas de buckets por fecha), puede trazar los buckets en varias dimensiones mediante las tres métricas que representan el X-axis (Eje X), el Y-axis (Eje Y) y el Size (Tamaño) de la burbuja.

5. Para encontrar los buckets que se han quedado inactivos, para el X-axis (Eje X), el Y-axis (Eje Y) y el Size (Tamaño), elija las métricas de Total storage (Almacenamiento total), % retrieval rate (Porcentaje de tasa de recuperación) y Average object size (Media de tamaño de los objetos).
6. En la sección Bubble analysis by buckets for date (Análisis de burbujas por buckets para fechas), busque buckets con tasas de recuperación cero (o cerca de cero) y un tamaño de almacenamiento relativo mayor y elija la burbuja que represente el bucket.

   Aparecerá un recuadro con opciones para obtener información más detallada. Haga una de las siguientes acciones:

   a. Para actualizar la pestaña Bucket para mostrar solo las métricas del bucket seleccionado, elija Drill down (Profundizar) y, a continuación, elija Apply (Aplicar).
   b. Para agregar los datos de nivel de bucket por cuenta, Región de AWS, clase de almacenamiento o bucket, elija Analyze by (Analizar por) y, a continuación, elija Dimension (Dimensión). Por ejemplo, para agregar por clase de almacenamiento, elija Storage class (Clase de almacenamiento) para Dimension (Dimensión).

   Para encontrar buckets que se han enfriado, haga un análisis de burbujas con las métricas de almacenamiento total, porcentaje de tasa de recuperación y tamaño promedio de objetos . Busque buckets con tasas de recuperación cero (o cerca de cero) y un tamaño de almacenamiento relativo mayor.

   La pestaña Bucket del panel se actualiza para mostrar los datos de la agregación o el filtro seleccionados. Si ha agregado por clase de almacenamiento u otra dimensión, esa nueva pestaña se abre en el panel (por ejemplo, la pestaña Storage class [Clase de almacenamiento]).

Paso 2: investigar buckets inactivos

Desde aquí, puede identificar a los propietarios de buckets inactivos en la cuenta u organización y averiguar si ese almacenamiento sigue siendo necesario. Después, para optimizar los costos, configure las opciones de vencimiento del ciclo de vida para esos buckets o archive los datos en una de las clases de almacenamiento de Amazon S3 Glacier.

Para evitar que avance el problema de los buckets inactivos, puede hacer la transición automática de los datos mediante las configuraciones de S3 Lifecycle para los buckets o habilitar el archivo automático con S3 Intelligent-Tiering.

También puede utilizar el paso 1 para identificar los buckets en caliente. A continuación, puede asegurarse de que estos buckets utilicen la S3 storage class (Clase de almacenamiento de S3) correcta para garantizar que atiendan las solicitudes de la manera más eficaz en términos de rendimiento y costo.

Localice cargas multiparte incompletas

Puede utilizar las cargas multiparte para cargar objetos muy grandes (hasta 5 TB) como un conjunto de piezas para mejorar el rendimiento y obtener una recuperación más rápida de los problemas de red. En los casos en que el proceso de carga multiparte no finaliza, las partes incompletas permanecen en el bucket (en un estado inutilizable). Estas partes incompletas generan gastos de almacenamiento hasta que finalice el proceso de carga o hasta que se eliminen las partes incompletas. Para obtener más información, consulte Carga y copia de objetos con la carga multiparte.

Con Lente de almacenamiento de S3, puede identificar el número de bytes de carga multiparte incompletos en la cuenta o en toda la organización, incluyendo cargas multiparte incompletas que tienen más de 7 días de antigüedad. Para ver una lista completa de métricas de carga multiparte incompletas, consulte Glosario de métricas de Amazon S3 Storage Lens.

Como práctica recomendada, recomendamos configurar las reglas del ciclo de vida para que venzan las cargas multiparte incompletas que tengan una antigüedad superior a un número específico de días. Cuando cree la regla de ciclo de vida para hacer que venzan las cargas multiparte incompletas, le recomendamos 7 días como buen punto de partida.

Paso 1: revisar las tendencias generales para las cargas multiparte incompletas

1. Inicie sesión en la AWS Management Console y abra la consola de Amazon S3 en https://console.aws.amazon.com/s3/.
2. En el panel de navegación izquierdo, elija Storage Lens, Dashboards (Paneles).
3. En la lista Dashboards (Paneles), elija el panel que desea ver.
4. En la sección Snapshot for date (Instantánea para fechas), en Metrics categories (Categorías de métricas), elija Cost optimization (Optimización de costos).

   La sección Snapshot for date (Instantánea para fechas) se actualiza para mostrar las métricas de Cost optimization (Optimización de costos), que incluyen Incomplete multipart upload bytes greater than 7 days old (Bytes de carga multiparte incompletos con más de 7 días de antigüedad).

En cualquier gráfico del panel de Lente de almacenamiento de S3, puede ver métricas de cargas multiparte incompletas. Puede utilizar estas métricas para evaluar más el impacto de los bytes de carga multiparte incompletos en el almacenamiento, incluida la contribución a las tendencias generales de crecimiento. También puede profundizar en los niveles de agregación más profundos mediante las pestañas de Account (Cuenta), Región de AWS, Bucket o Storage class (Clase de almacenamiento) para un análisis más profundo de los datos. Para ver un ejemplo, consulte Descubra buckets en frío de Amazon S3.

Paso 2: identificar los buckets que tengan la mayor cantidad de bytes de carga multiparte incompletos, pero que no tengan reglas de ciclo de vida para anular las cargas multiparte incompletas

Requisito previo

Para ver la métrica de Abort incomplete multipart upload lifecycle rule count (Abortar recuento de reglas del ciclo de vida de carga multiparte incompleta) en el panel de Lente de almacenamiento de S3, debe habilitar Advanced metrics and recommendations (Métricas y recomendaciones avanzadas) de Lente de almacenamiento de S3 y, a continuación, seleccionar las Advanced cost optimization metrics (Métricas de optimización de costos avanzadas). Para obtener más información, consulte Creación y actualización de los paneles de Amazon S3 Storage Lens.

1. Inicie sesión en la AWS Management Console y abra la consola de Amazon S3 en https://console.aws.amazon.com/s3/.
2. En el panel de navegación izquierdo, elija Storage Lens, Dashboards (Paneles).
3. En la lista Dashboards (Paneles), elija el panel que desea ver.
4. Para identificar buckets específicos que acumulen cargas multiparte incompletas con más de 7 días de antigüedad, vaya a la sección Top N overview for date (Información general de las N principales para fechas).

   De forma predeterminada, la sección Top N overview for date (Información general de las N principales para fechas) muestra las métricas de los 3 buckets principales. Puede aumentar o disminuir el número de buckets en el campo Top N (N principales). La sección Top N overview for date (Información general de las N principales para fechas) también muestra el cambio porcentual con respecto al día o la semana anteriores y un minigráfico para visualizar la tendencia. (Esta tendencia es una tendencia de 14 días para las métricas gratuitas y una tendencia de 30 días para las métricas y recomendaciones avanzadas).

> **Note**
>
> Con las métricas y recomendaciones avanzadas de Lente de almacenamiento de S3, las métricas están disponibles para consultas durante 15 meses. Para obtener más información, consulte Selección de métricas.

5. Para Metric (Métrica), elija Incomplete multipart upload bytes greater than 7 days old (Bytes de carga multiparte incompletos con más de 7 días de antigüedad) en la categoría Cost optimization (Optimización de costos).

   En Top number buckets (Buckets de número principales), puede ver los buckets con la mayoría de bytes de almacenamiento de carga multiparte incompletos que tienen más de 7 días de antigüedad.

6. Para ver las métricas de nivel de bucket más detalladas de las cargas multipartes incompletas, desplácese hasta la parte superior de la página y, a continuación, elija la pestaña Bucket.

7. Desplácese hacia abajo hasta la sección Buckets. Para Metrics categories (Categorías de métricas), seleccione Cost optimization (Optimización de costos). A continuación, elimine Summary (Resumen).

   La lista de Buckets se actualiza para mostrar todas las métricas de Cost optimization (Optimización de costos) disponibles para los buckets mostrados.

8. Para filtrar la lista de Buckets para mostrar solo métricas de optimización de costos específicas, elija el icono de preferencias ().

9. Desactive todas las métricas de optimización de costos hasta que solo permanezcan seleccionadas Incomplete multipart upload bytes greater than 7 days old (Bytes de carga multiparte incompleta de más de 7 días de antigüedad) y Abort incomplete multipart upload lifecycle rule count (Abortar el recuento de reglas de ciclo de vida de carga multiparte incompleta).

10. (Opcional) En Page size (Tamaño de página), elija el número de buckets que desea mostrar en la lista.

11. Elija Confirm (Confirmar).

    La lista de Buckets se actualiza para mostrar las métricas en el nivel de bucket para las cargas multiparte incompletas y los recuentos de reglas del ciclo de vida. Puede usar estos datos para

identificar los buckets que tengan la mayor cantidad de bytes de carga multiparte incompletos, que tengan más de 7 días de antigüedad y a los que les falten las reglas del ciclo de vida para abortar las cargas multiparte incompletas. A continuación, puede acceder a estos buckets en la consola de S3 y agregar reglas de ciclo de vida para eliminar las cargas multipartes incompletas abandonadas.

Paso 3: agregar una regla de ciclo de vida para eliminar las cargas multiparte incompletas transcurridos 7 días

Para administrar automáticamente las cargas multiparte incompletas, puede usar la consola de S3 para crear una política de ciclo de vida para que caduquen los bytes de carga multiparte incompletos de un bucket pasado un número concreto de días. Para obtener más información, consulte Configuración de una política de ciclo de vida del bucket para eliminar cargas multiparte incompletas.

Reduzca la cantidad de versiones no actuales retenidas

Cuando está habilitado, el control de versiones de S3 retiene varias copias del mismo objeto que se pueden utilizar para recuperar datos rápidamente si un objeto se elimina o sobrescribe accidentalmente. Si ha habilitado el control de versiones de S3 sin configurar las reglas del ciclo de vida para hacer la transición o que venzan las versiones no actuales, se puede acumular una gran cantidad de versiones anteriores no actuales, lo que puede tener implicaciones en los costos de almacenamiento. Para obtener más información, consulte Usar el control de versiones en buckets de S3.

Paso 1: identificar buckets con la mayoría de las versiones de objetos no actuales

1. Inicie sesión en la AWS Management Console y abra la consola de Amazon S3 en https://console.aws.amazon.com/s3/.
2. En el panel de navegación izquierdo, elija Storage Lens, Dashboards (Paneles).
3. En la lista Dashboards (Paneles), elija el panel que desea ver.
4. En la sección Snapshot for date (Instantánea para fechas), en Metric categories (Categorías de métricas), elija Cost optimization (Optimización de costos).

   La sección Snapshot for date (Instantánea para fechas) se actualiza para mostrar las métricas de Cost optimization (Optimización de costos), que incluyen la métrica del % noncurrent version bytes (Porcentaje de bytes de la versión no actual). La métrica del % noncurrent version bytes (Porcentaje de bytes de versión no actual) representa la proporción del total de bytes de

almacenamiento que se atribuye a versiones no actuales, dentro del alcance del panel y para la fecha seleccionada.

> **Note**
>
> Si el % noncurrent version bytes (Porcentaje de bytes de versiones no actuales) es superior al 10 % del almacenamiento a nivel de cuenta, es posible que se estén almacenando demasiadas versiones de objetos.

5. Para identificar buckets específicos que acumulan una gran cantidad de versiones no actuales:

   a. Desplácese hacia abajo hasta la sección Top N overview for date (Información general de N principales para fechas). Para Top N (N principales), ingrese el número de buckets de los que desearía ver los datos.

   b. Para Metric (Métrica), elija % noncurrent version bytes (Porcentaje de bytes de versión no actual).

      En Top number buckets (Buckets de número principales), puede ver los buckets (para el número que especificó) con el mayor % noncurrent version bytes (Porcentaje de bytes de versión no actuales). La sección Top N overview for date (Información general de las N principales para fechas) también muestra el cambio porcentual con respecto al día o la semana anteriores y un minigráfico para visualizar la tendencia. Esta tendencia es una tendencia de 14 días para las métricas gratuitas y una tendencia de 30 días para las métricas y recomendaciones avanzadas.

      > **Note**
      >
      > Con las métricas y recomendaciones avanzadas de Lente de almacenamiento de S3, las métricas están disponibles para consultas durante 15 meses. Para obtener más información, consulte Selección de métricas.

   c. Para ver métricas en el nivel de bucket más detalladas para las versiones de objetos no actuales, desplácese hasta la parte superior de la página y, a continuación, elija la pestaña Bucket.

      En cualquier gráfico o visualización del panel de Lente de almacenamiento de S3, puede profundizar en los niveles de agregación más profundos mediante las pestañas Account

(Cuenta), Región de AWS, Storage class (Clase de almacenamiento) o Bucket. Para ver un ejemplo, consulte Descubra buckets en frío de Amazon S3.

d. En la sección Buckets, para Metric categories (Categorías de métricas), seleccione Cost optimization (Optimización de costos). A continuación, elimine Summary (Resumen).

   Ahora puede ver la métrica de % noncurrent version bytes (Porcentaje de bytes de versión no actual), junto con otras métricas relacionadas con las versiones no actuales.

Paso 2: identificar los buckets a los que les faltan las reglas del ciclo de vida de transición y vencimiento para administrar las versiones no actuales

Requisito previo

Para ver las métricas de Noncurrent version transition lifecycle rule count (Recuento de reglas del ciclo de vida de transición de versiones no actuales) y de Noncurrent version expiration lifecycle rule count (Recuento de reglas del ciclo de vida de vencimiento de versiones no actuales) en el panel de Lente de almacenamiento de S3, debe habilitar las Advanced metrics and recommendations (Recomendaciones y métricas avanzadas) de Lente de almacenamiento de S3 y, a continuación, seleccionar Advanced cost optimization metrics (Métricas de optimización de costos avanzadas). Para obtener más información, consulte Creación y actualización de los paneles de Amazon S3 Storage Lens.

1. Inicie sesión en la AWS Management Console y abra la consola de Amazon S3 en https://console.aws.amazon.com/s3/.
2. En el panel de navegación izquierdo, elija Storage Lens, Dashboards (Paneles).
3. En la lista Dashboards (Paneles), elija el panel que desea ver.
4. En el panel de Storage Lens, elija la pestaña Bucket.
5. Desplácese hacia abajo hasta la sección Buckets. Para Metrics categories (Categorías de métricas), seleccione Cost optimization (Optimización de costos). A continuación, elimine Summary (Resumen).

   La lista de Buckets se actualiza para mostrar todas las métricas de Cost optimization (Optimización de costos) disponibles para los buckets mostrados.
6. Para filtrar la lista de Buckets para mostrar solo métricas de optimización de costos específicas, elija el icono de preferencias ().

7. Desactive los botones de todas las métricas de optimización de costos hasta que solo queden seleccionadas las siguientes:

   - % noncurrent version bytes (Porcentaje de bytes de la versión que no es actual)
   - Noncurrent version transition lifecycle rule count (Recuento de reglas del ciclo de vida de transición de versiones no actuales)
   - Noncurrent version expiration lifecycle rule count (Recuento de reglas del ciclo de vida de vencimiento de versiones no actuales)

8. (Opcional) En Page size (Tamaño de página), elija el número de buckets que desea mostrar en la lista.

9. Elija Confirm (Confirmar).

   La lista de buckets se actualiza para mostrar las métricas de los bytes de las versiones no actuales y los recuentos de reglas del ciclo de vida de las versiones no actuales. Puede usar estos datos para identificar los buckets que tienen un alto porcentaje de bytes de versiones no actuales, pero a los que les faltan las reglas del ciclo de vida de transición y vencimiento. A continuación, puede navegar hasta estos buckets en la consola de S3 y agregarles reglas del ciclo de vida.

Paso 3: agregar reglas del ciclo de vida para hacer la transición o hacer que venzan las versiones de objetos no actuales

Después de que haya determinar los buckets que requieren más investigación, puede navegar hasta los buckets dentro de la consola de S3 y agregar una regla de ciclo de vida para que venzan las versiones no actuales después de una cantidad específica de días. De manera opcional, para reducir los costos mientras se retienen las versiones no actuales, puede configurar una regla del ciclo de vida para pasar versiones no actuales a una de las clases de almacenamiento de Amazon S3 Glacier. Para obtener más información, consulte Ejemplo 6: especificar una regla del ciclo de vida para un bucket habilitado para el control de versiones.

Identifique los buckets que no tienen reglas de ciclo de vida y revise el recuento de reglas del ciclo de vida

Lente de almacenamiento de S3 proporciona métricas de recuento de reglas del ciclo de vida de S3 que puede utilizar para identificar los buckets a los que les faltan reglas del ciclo de vida. Para encontrar buckets que no tengan reglas del ciclo de vida, puede utilizar la métrica de Total buckets without lifecycle rules (Buckets totales sin reglas del ciclo de vida). Es posible que un bucket sin la

configuración del ciclo de vida de S3 tenga un almacenamiento que ya no necesite o que pueda migrar a una clase de almacenamiento de menor costo. También puede utilizar las métricas del recuento de reglas del ciclo de vida para identificar los buckets a los que les faltan tipos específicos de reglas del ciclo de vida, como las reglas de vencimiento o transición.

Requisito previo

Para ver las métricas de recuento de reglas del ciclo de vida y la métrica de Total buckets without lifecycle rules (Buckets totales sin reglas del ciclo de vida) en el panel de Lente de almacenamiento de S3, debe habilitar las Advanced metrics and recommendations (Métricas y recomendaciones avanzadas) de Lente de almacenamiento de S3 y, a continuación, seleccionar Advanced cost optimization metrics (Métricas de optimización de costos avanzadas). Para obtener más información, consulte Creación y actualización de los paneles de Amazon S3 Storage Lens.

Paso 1: identificar buckets sin reglas del ciclo de vida

1. Inicie sesión en la AWS Management Console y abra la consola de Amazon S3 en https://console.aws.amazon.com/s3/.
2. En el panel de navegación izquierdo, elija Storage Lens, Dashboards (Paneles).
3. En la lista Dashboards (Paneles), elija el panel que desea ver.
4. Para identificar buckets específicos sin reglas del ciclo de vida, desplácese hacia abajo hasta la sección Top N overview for date (Información general de las N principales para fechas).

   De forma predeterminada, la sección Top N overview for date (Información general de las N principales para fechas) muestra las métricas de los 3 buckets principales. En el campo Top N (N principales), puede aumentar el número de buckets. La sección Top N overview for date (Información general de las N principales para fechas) también muestra el cambio porcentual con respecto al día o la semana anteriores y un minigráfico para visualizar la tendencia. Esta tendencia es una tendencia de 14 días para las métricas gratuitas y una tendencia de 30 días para las métricas y recomendaciones avanzadas.

   > **Note**
   >
   > Con las métricas y recomendaciones avanzadas de Lente de almacenamiento de S3, las métricas están disponibles para consultas durante 15 meses. Para obtener más información, consulte Selección de métricas.

5. Para Metric (Métrica), elija Total buckets without lifecycle rules (Total de buckets sin reglas del ciclo de vida) en la categoría Cost optimization (Optimización de costos).
6. Revise los siguientes datos para Total buckets without lifecycle rules (Total de buckets sin reglas del ciclo de vida):

   - Top number accounts (Cuentas de números principales): consulte qué cuentas tienen más buckets sin reglas del ciclo de vida.
   - Top number Regions (Regiones de números principales): consulte un desglose de los buckets sin reglas del ciclo de vida por región.
   - Top number buckets (Buckets de números principales): consulte qué buckets no tienen reglas del ciclo de vida.

   En cualquier gráfico o visualización del panel de Lente de almacenamiento de S3, puede profundizar en los niveles de agregación más profundos mediante las pestañas Account (Cuenta), Región de AWS, Storage class (Clase de almacenamiento) o Bucket. Para ver un ejemplo, consulte Descubra buckets en frío de Amazon S3.

   Después de identificar qué buckets no tienen reglas del ciclo de vida, también puede revisar los recuentos de reglas del ciclo de vida específicos para los buckets.

Paso 2: revisar el recuento de reglas del ciclo de vida para los buckets

1. Inicie sesión en la AWS Management Console y abra la consola de Amazon S3 en https://console.aws.amazon.com/s3/.
2. En el panel de navegación izquierdo, elija Storage Lens, Dashboards (Paneles).
3. En la lista Dashboards (Paneles), elija el panel que desea ver.
4. En el panel de Lente de almacenamiento de S3, elija la pestaña Bucket.
5. Desplácese hacia abajo hasta la sección Buckets. En Metrics categories (Categorías de métricas), seleccione Cost optimization (Optimización de costos). A continuación, elimine Summary (Resumen).

   La lista de Buckets se actualiza para mostrar todas las métricas de Cost optimization (Optimización de costos) disponibles para los buckets mostrados.

6. Para filtrar la lista de Buckets para mostrar solo métricas de optimización de costos específicas, elija el icono de preferencias ().
7. Desactive los botones de todas las métricas de optimización de costos hasta que solo queden seleccionadas las siguientes:
   - Transition lifecycle rule count (Recuento de reglas del ciclo de vida de la transición)
   - Expiration lifecycle rule count (Recuento de reglas del ciclo de vida del vencimiento)
   - Noncurrent version transition lifecycle rule count (Recuento de reglas de ciclo de vida de transición de versiones no actuales)
   - Noncurrent version expiration lifecycle rule count (Recuento de reglas del ciclo de vida de vencimiento de versiones no actuales)
   - Abort incomplete multipart upload lifecycle rule count (Abortar el recuento de reglas del ciclo de vida de carga multiparte incompleto)
   - Total lifecycle rule count (Recuento de reglas del ciclo de vida total)
8. (Opcional) En Page size (Tamaño de página), elija el número de buckets que desea mostrar en la lista.
9. Elija Confirm (Confirmar).

   La lista de Buckets se actualiza para mostrar las métricas del recuento de reglas del ciclo de vida para los buckets. Puede utilizar estos datos para identificar los buckets sin reglas del ciclo de vida o los buckets a los que les faltan tipos específicos de reglas del ciclo de vida, por ejemplo, reglas de vencimiento o transición. A continuación, puede navegar hasta estos buckets en la consola de S3 y agregarles reglas del ciclo de vida.

Paso 3: agregar reglas del ciclo de vida

Una vez que haya identificado los buckets sin reglas del ciclo de vida, puede agregar reglas del ciclo de vida. Para obtener más información, consulte Configuración de un ciclo de vida en un bucket y Ejemplos de configuración de S3 Lifecycle.

Uso de S3 Storage Lens para proteger sus datos

Puede utilizar las métricas de protección de datos de Lente de almacenamiento de Amazon S3 para identificar los buckets en los que no se han aplicado las prácticas recomendadas de protección de datos. Puede usar estas métricas para tomar medidas y aplicar una configuración estándar que

se ajuste a las prácticas recomendadas para proteger los datos en todos los buckets de la cuenta u organización. Por ejemplo, puede usar las métricas de protección de datos para identificar los buckets que no utilizan claves AWS Key Management Service (AWS KMS) (SSE-KMS) para el cifrado predeterminado o las solicitudes que utilizan AWS Signature Version 2 (SigV2).

Los siguientes casos de uso proporcionan estrategias para usar el panel de Lente de almacenamiento de S3 para identificar los valores atípicos y aplicar las prácticas recomendadas de protección de datos en todos los buckets de S3.

Temas

Identificar los buckets que no utilizan el cifrado del lado del servidor con AWS KMS para el cifrado predeterminado (SSE-KMS)

Con el cifrado predeterminado de Amazon S3, puede establecer el comportamiento de cifrado predeterminado para un bucket de S3. Para obtener más información, consulte the section called “Establecer el cifrado predeterminado de un bucket”.

Puede utilizar el SSE-KMS enabled bucket count (Recuento de buckets habilitados para SSE-KMS) y las métricas de % SSE-KMS enabled buckets (Porcentaje de buckets habilitados para SSE-KMS) para identificar los buckets que utilizan el cifrado del lado del servidor con claves AWS KMS (SSE-KMS) como cifrado predeterminado. Lente de almacenamiento de S3 también proporciona métricas para bytes no cifrados, objetos no cifrados, bytes cifrados y objetos cifrados. Para obtener una lista completa de métricas, consulte Glosario de métricas de Amazon S3 Storage Lens.

Puede analizar las métricas de cifrado de SSE-KMS en el contexto de las métricas de cifrado generales para identificar los buckets que no utilizan SSE-KMS. Si quiere usar SSE-KMS para todos los buckets de la cuenta u organización, puede actualizar la configuración de cifrado predeterminada para estos buckets para usar SSE-KMS. Además de SSE-KMS, puede utilizar el cifrado del lado del servidor con claves administradas de Amazon S3 (SSE-S3) o claves proporcionadas por el cliente (SSE-C). Para obtener más información, consulte Protección de los datos mediante el cifrado.

Paso 1: identificar qué buckets utilizan SSE-KMS para el cifrado predeterminado

1. Inicie sesión en la AWS Management Console y abra la consola de Amazon S3 en https://console.aws.amazon.com/s3/.
2. En el panel de navegación izquierdo, elija Storage Lens, Dashboards (Paneles).
3. En la lista Dashboards (Paneles), elija el nombre del panel que desea ver.
4. En la sección Trends and distributions (Tendencias y distribuciones), elija % SSE-KMS enabled bucket count (Porcentaje de recuento de buckets habilitado de SSE-KMS) para la métrica principal y % encrypted bytes (Porcentaje de bytes cifrados) para la métrica secundaria.

   El gráfico de Trend for date (Tendencias para fechas) se actualiza para mostrar las tendencias para SSE-KMS y bytes cifrados.
5. Para ver información más detallada e información en el nivel de bucket para SSE-KMS:

   a. Elija un punto del gráfico. Aparecerá un recuadro con opciones para obtener información más detallada.
   b. Elija la dimensión Buckets. A continuación, elija Apply (Aplicar).
6. En el gráfico Distribution by buckets for date (Distribución por buckets para fecha), elija la métrica SSE-KMS enabled bucket count (Recuento de buckets habilitado de SSE-KMS).
7. Ahora puede ver qué buckets tienen habilitado SSE-KMS y cuáles no.

Paso 2: actualizar la configuración de cifrado predeterminada del bucket

Ahora que ha determinado qué buckets utilizan SSE-KMS en el contexto del % encrypted bytes (Porcentaje de bytes cifrados), puede identificar los buckets que no utilizan SSE-KMS. A continuación, puede acceder opcionalmente a estos buckets en la consola de S3 y actualizar la configuración de cifrado predeterminada para usar SSE-KMS o SSE-S3. Para obtener más información, consulte Configuración del cifrado predeterminado.

Identificar los buckets que tienen habilitado el control de versiones de S3

Cuando está habilitada, la característica de control de versiones de S3 conserva varias versiones del mismo objeto que se pueden utilizar para recuperar datos rápidamente si un objeto se elimina o sobrescribe accidentalmente. Puede usar la métrica de Versioning-enabled bucket count (Recuento de buckets con control de versiones habilitado) para ver qué buckets utilizan el control de versiones de S3. A continuación, puede realizar una acción en la consola de S3 para habilitar el control de versiones de S3 para otros buckets.

Paso 1: identificar los buckets que tienen habilitado el control de versiones de S3

1. Inicie sesión en la AWS Management Console y abra la consola de Amazon S3 en https://console.aws.amazon.com/s3/.
2. En el panel de navegación, elija Storage Lens, Dashboards (Paneles).
3. En la lista Dashboards (Paneles), elija el nombre del panel que desea ver.
4. En la sección Trends and distributions (Tendencias y distribuciones), elija Versioning-enabled bucket count (Recuento de buckets con control de versiones habilitado) para la métrica principal y Buckets para la métrica secundaria.

   El gráfico de Trend for date (Tendencias para fechas) se actualiza para mostrar las tendencias para los buckets habilitados de control de versiones de S3. Justo debajo de la línea de tendencias, puede ver las subsecciones Storage class distribution (Distribución de clases de almacenamiento) y Region distribution (Distribución de regiones).
5. Para ver información más detallada sobre cualquiera de los buckets que se ven en el gráfico Trend for date (Tendencias para fechas) y poder realizar un análisis más profundo, haga lo siguiente:

   a. Elija un punto del gráfico. Aparecerá un recuadro con opciones para obtener información más detallada.

   b. Elija una dimensión para aplicarla a los datos para un análisis más profundo: Account (Cuenta), Región de AWS, Storage class (Clase de almacenamiento) o Bucket. A continuación, elija Apply (Aplicar).
6. En la sección Bubble analysis by buckets for date (Análisis de burbujas por buckets para fechas), elija las métricas Versioning-enabled bucket count (Recuento de buckets con el control de versiones habilitado), Buckets y Active buckets (Buckets activos).

   La sección Bubble analysis by buckets for date (Análisis de burbujas por buckets para fechas) se actualiza para mostrar los datos de las métricas que seleccionó. Puede usar estos datos para ver qué buckets tienen habilitado el control de versiones de S3 en el contexto del recuento total de buckets. En la sección Bubble analysis by buckets for date (Análisis de burbujas de buckets por fecha), puede trazar los buckets en varias dimensiones mediante las tres métricas que representan el X-axis (Eje X), el Y-axis (Eje Y) y el Size (Tamaño) de la burbuja.

Paso 2: habilitar el control de versiones de S3

Una vez que haya identificado los buckets que tienen habilitado el control de versiones de S3, podrá identificar los buckets que nunca lo han tenido habilitado o que tienen el control de versiones suspendido. A continuación, puede habilitar opcionalmente el control de versiones para estos buckets en la consola de S3. Para obtener más información, consulte Habilitar el control de versiones en buckets.

Identificar solicitudes que usen Signature Version 2 (SigV2) de AWS

Puede utilizar la métrica All unsupported signature requests (Todas las solicitudes de firma no admitidas) para identificar las solicitudes que utilizan AWS Signature Version 2 (SigV2). Estos datos le pueden ayudar a identificar aplicaciones específicas que utilizan SigV2. A continuación, puede migrar estas aplicaciones a AWS Signature Version 4 (SigV4).

SigV4 es el método de firma recomendado para todas las aplicaciones nuevas de S3. SigV4 proporciona una seguridad mejorada y es compatible con todas las Regiones de AWS. Para obtener más información, consulte Actualización de Amazon S3: período de desaprobación de Sigv2 extendido y modificado.

Requisito previo

Para ver All unsupported signature requests (Todas las solicitudes de firma no admitidas) en el panel de Lente de almacenamiento de S3, debe habilitar las Advanced metrics and recommendations (Métricas y recomendaciones avanzadas) de Lente de almacenamiento de S3 y, a continuación, seleccionar las Advanced data protection metrics (Métricas avanzadas de protección de datos). Para obtener más información, consulte Creación y actualización de los paneles de Amazon S3 Storage Lens.

Paso 1: examinar las tendencias de firma de SigV2 por Cuenta de AWS, región y bucket

1. Inicie sesión en la AWS Management Console y abra la consola de Amazon S3 en https://console.aws.amazon.com/s3/.
2. En el panel de navegación izquierdo, elija Storage Lens, Dashboards (Paneles).
3. En la lista Dashboards (Paneles), elija el nombre del panel que desea ver.
4. Para identificar buckets, cuentas y regiones específicos con solicitudes que utilizan SigV2:

    a. En Top N overview for date (Información general de las N principales para fechas), en Top N (N principales), ingrese el número de buckets de los que le gustaría ver los datos.

b. Para Metric (Métrica), elija All unsupported signature requests (Todas las solicitudes de firma no admitidas) en la categoría Data protection (Protección de datos).

La Top N overview for date (Información general de las N principales para fechas) se actualiza para mostrar datos para solicitudes SigV2 por cuenta, Región de AWS y bucket. La sección Top N overview for date (Información general de las N principales para fechas) también muestra el cambio porcentual con respecto al día o la semana anteriores y un minigráfico para visualizar la tendencia. Esta tendencia es una tendencia de 14 días para las métricas gratuitas y una tendencia de 30 días para las métricas y recomendaciones avanzadas.

> **Note**
>
> Con las métricas y recomendaciones avanzadas de Lente de almacenamiento de S3, las métricas están disponibles para consultas durante 15 meses. Para obtener más información, consulte Selección de métricas.

Paso 2: identificar los buckets a los que acceden las aplicaciones mediante solicitudes SigV2

1. Inicie sesión en la AWS Management Console y abra la consola de Amazon S3 en https://console.aws.amazon.com/s3/.
2. En el panel de navegación izquierdo, elija Storage Lens, Dashboards (Paneles).
3. En la lista Dashboards (Paneles), elija el nombre del panel que desea ver.
4. En el panel de Storage Lens, elija la pestaña Bucket.
5. Desplácese hacia abajo hasta la sección Buckets. En Metric categories (Categorías de métricas), elija Data protection (Protección de datos). A continuación, elimine Summary (Resumen).

   La lista de Buckets se actualiza para mostrar todas las métricas de Data protection (Protección de datos) disponibles para los buckets que se muestran.
6. Para filtrar la lista de Buckets para mostrar solo métricas de protección de datos específicas, elija el icono de preferencias ().
7. Desactive los botones de todas las métricas de protección de datos hasta que solo queden seleccionadas las siguientes métricas:

- All unsupported signature requests (Todas las solicitudes de firma no admitidas)
- % all unsupported signature requests (Porcentaje de todas las solicitudes de firma no admitidas)

8. (Opcional) En Page size (Tamaño de página), elija el número de buckets que desea mostrar en la lista.
9. Elija Confirm (Confirmar).

   La lista de Buckets se actualiza para mostrar las métricas en el nivel de bucket de las solicitudes SigV2. Puede usar estos datos para identificar buckets específicos que tienen solicitudes SigV2. A continuación, puede utilizar esta información para migrar las aplicaciones a SigV4. Para obtener más información, consulte Autenticación de solicitudes (AWS Signature Version 4) en la Referencia de la API de Amazon Simple Storage Service.

Cuente el número total de reglas de replicación para cada bucket

Con la Replicación de S3 es posible copiar objetos entre buckets de Amazon S3 de forma automática y asincrónica. Los buckets que están configurados para reproducción de objetos pueden pertenecer a la misma Cuenta de AWS o a cuentas diferentes. Para obtener más información, consulte Información general de la replicación de objetos.

Puede utilizar las métricas de recuento de reglas de Replicación de Lente de almacenamiento de S3 para obtener información detallada por bucket sobre los buckets configurados para la replicación. Esta información incluye reglas de replicación dentro y entre buckets y regiones.

Requisito previo

Para ver métricas de recuento de reglas de replicación en el panel de Lente de almacenamiento de S3, debe habilitar Lente de almacenamiento de S3 Advanced metrics and recommendations (Métricas y recomendaciones avanzadas) y, a continuación, seleccionar Advanced data protection metrics (Métricas de protección de datos avanzadas). Para obtener más información, consulte Creación y actualización de los paneles de Amazon S3 Storage Lens.

Paso 1: cuente el número total de reglas de replicación para cada bucket

1. Inicie sesión en la AWS Management Console y abra la consola de Amazon S3 en https://console.aws.amazon.com/s3/.
2. En el panel de navegación izquierdo, elija Storage Lens, Dashboards (Paneles).

3. En la lista Dashboards (Paneles), elija el nombre del panel que desea ver.
4. En el panel de Storage Lens, elija la pestaña Bucket.
5. Desplácese hacia abajo hasta la sección Buckets. En Metric categories (Categorías de métricas), elija Data protection (Protección de datos). A continuación, elimine Summary (Resumen).
6. Para filtrar la lista de buckets para mostrar solo las métricas del recuento de reglas de replicación, elija el icono de preferencias ().
7. Desactive los botones de todas las métricas de protección de datos hasta que solo queden seleccionadas las métricas de recuento de reglas de replicación:
   - Same-Region Replication rule count (Recuento de reglas de replicación de la misma región)
   - Cross-Region Replication rule count (Recuento de reglas de replicación entre regiones)
   - Same-account replication rule count (Recuento de reglas de replicación de la misma cuenta)
   - Cross-account replication rule count (Recuento de reglas de replicación entre cuentas)
   - Total replication rule count (Recuento de reglas de replicación total)
8. (Opcional) En Page size (Tamaño de página), elija el número de buckets que desea mostrar en la lista.
9. Elija Confirm (Confirmar).

Paso 2: agregar reglas de replicación

Una vez que disponga de un recuento de reglas de replicación por bucket, puede crear reglas de replicación adicionales si lo desea. Para obtener más información, consulte Ejemplos para configurar la replicación en directo.

Identificar el porcentaje de bytes de Object Lock

Con S3 Object Lock, puede almacenar objetos con un modelo de escritura única y lectura múltiple (WORM). Puede usar Object Lock para ayudarle a evitar que se eliminen o se sobrescriban objetos durante un periodo de tiempo determinado o de manera indefinida. Puede habilitar Object Lock solo al crear un bucket y también habilitar el control de versiones de S3. Sin embargo, puede editar el periodo de retención de las versiones individuales de los objetos o aplicar retenciones legales a los buckets que tengan habilitado Object Lock. Para obtener más información, consulte Usar Bloqueo de objetos de S3.

Puede usar las métricas de Object Lock en Lente de almacenamiento de S3 para ver la métrica % Object Lock bytes (Porcentaje de bytes de Object Lock) de la cuenta u organización. Puede usar esta información para identificar los buckets de la cuenta u organización que no siguen las prácticas recomendadas de protección de datos.

1. Inicie sesión en la AWS Management Console y abra la consola de Amazon S3 en https://console.aws.amazon.com/s3/.
2. En el panel de navegación izquierdo, elija Storage Lens, Dashboards (Paneles).
3. En la lista Dashboards (Paneles), elija el nombre del panel que desea ver.
4. En la sección Snapshot (Instantánea), en Metrics categories (Categorías de métricas), elija Data protection (Protección de datos).

   La sección Snapshot (Instantánea) se actualiza para mostrar las métricas de protección de datos, incluida la métrica % Object Lock bytes (Porcentaje de bytes de Object Lock). Puede ver el porcentaje total de bytes de Object Lock de la cuenta u organización.
5. Para ver el % Object Lock bytes (Porcentaje de bytes de Object Lock) por bucket, desplácese hacia abajo hasta la sección Top N overview (Información general de las N principales).

   Para obtener datos en el nivel de objeto para Object Lock, también puede usar las métricas de Object Lock object count (Recuento de objetos de Object Lock) y % Object Lock objects (Porcentaje de objetos de Object Lock).
6. Para Metric (Métrica), elija % Object Lock bytes (Porcentaje de bytes de Object Lock) de la categoría Data protection (Protección de datos).

   De forma predeterminada, la sección Top N overview for date (Información general de las N principales para fechas) muestra las métricas de los 3 buckets principales. En el campo Top N (N principales), puede aumentar el número de buckets. La sección Top N overview for date (Información general de las N principales para fechas) también muestra el cambio porcentual con respecto al día o la semana anteriores y un minigráfico para visualizar la tendencia. Esta tendencia es una tendencia de 14 días para las métricas gratuitas y una tendencia de 30 días para las métricas y recomendaciones avanzadas.

> **Note**
>
> Con las métricas y recomendaciones avanzadas de Lente de almacenamiento de S3, las métricas están disponibles para consultas durante 15 meses. Para obtener más información, consulte Selección de métricas.

7. Revise los siguientes datos para ver % Object Lock bytes (Porcentaje de bytes de Object Lock):

   - Top number accounts (Cuentas de números principales): consulte qué cuentas tienen el % Object Lock bytes (Porcentaje de bytes de Object Lock) más alto y más bajo.
   - Top number Regions (Regiones de números principales): consulte un desglose de % Object Lock bytes (Porcentaje de bytes de Object Lock) por región.
   - Top number buckets (Buckets de números principales): consulte qué buckets tienen el % Object Lock bytes (Porcentaje de bytes de Object Lock) más alto y más bajo.

Uso de S3 Storage Lens para auditar la configuración de la propiedad de objetos

La propiedad de objetos de Amazon S3 es una configuración en el nivel de bucket de S3 que puede utilizar para desactivar las listas de control de acceso (ACL) y controlar la propiedad de los objetos del bucket. Si establece la propiedad de objetos en propietario del bucket obligatorio, puede desactivar las listas de control de acceso (ACL) y tomar posesión de cada objeto del bucket. Este enfoque simplifica la administración de acceso para los datos almacenados en Amazon S3.

De forma predeterminada, cuando otra Cuenta de AWS carga un objeto en el bucket de S3, esa cuenta (el escritor del objeto) es propietario del objeto, tiene acceso a él y puede conceder acceso a él a otros usuarios a través de ACL. Puede utilizar la propiedad de objetos para cambiar este comportamiento predeterminado.

La mayoría de los casos de uso modernos de Amazon S3 ya no requieren el uso de ACL. Por lo tanto, le recomendamos desactivar las ACL, excepto en circunstancias inusuales en las que necesite controlar el acceso de cada objeto de manera individual. Al establecer la propiedad de objetos en propietario del bucket obligatorio, puede desactivar las ACL y confiar en políticas para el control de acceso. Para obtener más información, consulte Control de la propiedad de los objetos y desactivación de las ACL del bucket.

Con las métricas de administración de acceso de S3 Storage Lens, puede identificar los buckets que no tienen las ACL deshabilitadas. Tras identificar estos buckets, puede migrar los permisos de ACL a las políticas y desactivar las ACL para estos buckets.

Temas

Paso 1: identificar las tendencias generales de la configuración de propiedad de objetos

1. Inicie sesión en la AWS Management Console y abra la consola de Amazon S3 en https://console.aws.amazon.com/s3/.
2. En el panel de navegación izquierdo, elija Storage Lens, Dashboards (Paneles).
3. En la lista Dashboards (Paneles), elija el nombre del panel que desea ver.
4. En la sección Snapshot for date (Instantánea para fecha), en Metrics categories (Categorías de métricas), elija Access management (Administración de acceso).

   La sección Snapshot for date (Instantánea para fecha) se actualiza para mostrar la métrica de % Object Ownership bucket owner enforced (Porcentaje de propiedad de objetos del propietario del bucket obligatorio). Puede ver el porcentaje total de buckets de la cuenta u organización que utilizan la configuración de propietario del bucket obligatorio de propiedad de objetos para desactivar las ACL.

Paso 2: identificar las tendencias en el nivel de bucket de la configuración de propiedad de objetos

1. Inicie sesión en la AWS Management Console y abra la consola de Amazon S3 en https://console.aws.amazon.com/s3/.
2. En el panel de navegación izquierdo, elija Storage Lens, Dashboards (Paneles).
3. En la lista Dashboards (Paneles), elija el nombre del panel que desea ver.
4. Para ver métricas más detalladas de nivel de bucket, elija la pestaña Bucket.
5. En la sección Distribution by buckets for date (Distribución de buckets por fecha), elija la métrica de % Object Ownership bucket owner enforced (Porcentaje de propiedad de objetos del propietario del bucket obligatorio).

El gráfico se actualiza para mostrar un desglose por bucket para % Object Ownership bucket owner enforced (Porcentaje de propiedad de objetos del propietario del bucket obligatorio). Puede ver qué buckets utilizan la configuración de propietario del bucket obligatorio de propiedad de objetos para desactivar las ACL.

6. Para ver la configuración de propietario del bucket obligatorio en contexto, desplácese hacia abajo hasta la sección Buckets. Para las Metrics categories (Categorías de las métricas), seleccione Access management (Administración de acceso). A continuación, elimine Summary (Resumen).

   La lista de Buckets muestra los datos de las tres configuraciones de propiedad de objetos: propietario de bucket obligatorio, propietario de bucket preferido y escritor de objetos.

7. Para filtrar la lista de Buckets para mostrar solo métricas de una configuración de propiedad de objetos específica, elija el icono de preferencias ().

8. Elimine las métricas que no desee ver.

9. (Opcional) En Page size (Tamaño de página), elija el número de buckets que desea mostrar en la lista.

10. Elija Confirm.

Paso 3: actualice la configuración de propiedad de objetos a propietario del bucket obligatorio para desactivar las ACL

Una vez que haya identificado los buckets que utilizan la configuración preferida del escritor de objetos y el propietario del bucket para la propiedad de objetos, puede migrar los permisos de ACL a políticas de bucket. Cuando haya terminado de migrar los permisos de ACL, puede actualizar la configuración de propiedad de objetos a propietario del bucket obligatorio para desactivar las ACL. Para obtener más información, consulte Requisitos previos para desactivar las ACL.

Uso de métricas de S3 Storage Lens para mejorar el rendimiento

Si tiene la opción S3 Storage Lens advanced metrics (Métricas avanzadas de S3 Storage Lens) habilitada, puede usar las métricas de código de estado detalladas para obtener recuentos para solicitudes exitosas o erróneas. Puede utilizar esta información para solucionar problemas de acceso o problemas de rendimiento. Las métricas de código de estado detalladas muestran los recuentos para los códigos de estado HTTP, como 403 Forbidden (Prohibido) y 503 Service Unavailable

(Servicio no disponible). Puede examinar las tendencias generales para obtener métricas de código de estado detalladas en los buckets, cuentas y organizaciones de S3. A continuación, puede analizar en detalle las métricas en el nivel de bucket para identificar las cargas de trabajo que actualmente acceden a estos buckets y que están causando errores.

Por ejemplo, puede consultar la métrica 403 Forbidden error count (Recuento de error 403 Forbidden [Prohibido]) para identificar las cargas de trabajo que acceden a los buckets sin haberse aplicado los permisos correctos. Una vez que haya identificado estas cargas de trabajo, puede analizar en profundidad fuera de S3 Storage Lens para solucionar los errores de 403 Forbidden (Prohibido).

En este ejemplo, se muestra cómo realizar un análisis de tendencias para el error 403 Forbidden (Prohibido) mediante el 403 Forbidden error count (Recuento de error 403 Forbidden) y las métricas de % 403 Forbidden errors (Porcentaje de errores 403 Forbidden [Prohibido]). Puede usar estas métricas para identificar las cargas de trabajo que acceden a los buckets sin aplicar los permisos correctos. Puede realizar un análisis de tendencias similar para cualquiera de las demás Detailed status code metrics (Métricas de código de estado detalladas). Para obtener más información, consulte Glosario de métricas de Amazon S3 Storage Lens.

Requisito previo

Para ver Detailed status code metrics (Métricas de código de estado detalladas) en el panel de S3 Storage Lens, debe habilitar las Advanced metrics and recommendations (Métricas y recomendaciones avanzadas) de S3 Storage Lens y, a continuación, seleccionar Detailed status code metrics (Métricas de código de estado detalladas). Para obtener más información, consulte Creación y actualización de los paneles de Amazon S3 Storage Lens.

Temas

- Paso 1: realizar un análisis de tendencias para un código de estado HTTP individual
- Paso 2: analizar los recuentos de errores por bucket
- Paso 3: solucionar los errores

Paso 1: realizar un análisis de tendencias para un código de estado HTTP individual

1. Inicie sesión en la AWS Management Console y abra la consola de Amazon S3 en https://console.aws.amazon.com/s3/.
2. En el panel de navegación izquierdo, elija Storage Lens, Dashboards (Paneles).
3. En la lista Dashboards (Paneles), elija el nombre del panel que desea ver.

4. En la sección Trends and distributions (Tendencias y distribuciones), en Primary metric (Métrica principal), elija 403 Forbidden error count (Recuento de error 403 Forbidden [Prohibido]) en la categoría Detailed status codes (Códigos de estado detallados). Para la Secondary metric (Métrica secundaria), elija % 403 Forbidden errors (Porcentaje de errores 403 Forbidden [Prohibido]).
5. Desplácese hacia abajo hasta la sección Top N overview for date (Información general de N principales para fechas). Para Metrics (Métricas), elija 403 Forbidden error count (Recuento de error 403 Forbidden [Prohibido]) o % 403 Forbidden errors (Porcentaje de errores 403 Forbidden [Prohibido]) de la categoría Detailed status codes (Códigos de estado detallados).

   La sección Top N overview for date (Información general de N principales) se actualiza para mostrar los recuentos de errores 403 Forbidden (prohibido) principales por cuenta, Región de AWS y bucket.

Paso 2: analizar los recuentos de errores por bucket

1. Inicie sesión en la AWS Management Console y abra la consola de Amazon S3 en https://console.aws.amazon.com/s3/.
2. En el panel de navegación izquierdo, elija Storage Lens, Dashboards (Paneles).
3. En la lista Dashboards (Paneles), elija el nombre del panel que desea ver.
4. En el panel de Storage Lens, elija la pestaña Bucket.
5. Desplácese hacia abajo hasta la sección Buckets. Para Metrics categories (Categorías de métricas), seleccione métricas de Detailed status code (Código de estado detallado). A continuación, elimine Summary (Resumen).

   La lista de buckets se actualiza para mostrar todas las métricas detalladas del código de estado disponibles. Puede usar esta información para ver qué buckets tienen una gran proporción de determinados códigos de estado HTTP y qué códigos de estado son comunes en todos los buckets.
6. Para filtrar la lista de Buckets para mostrar solo métricas de código de estado detalladas específicas, elija el icono de preferencias ().
7. Desactive los botones para ver las métricas de código de estado detalladas que no desee ver en la lista de Buckets.

8. (Opcional) En Page size (Tamaño de página), elija el número de buckets que desea mostrar en la lista.
9. Elija Confirm.

   La lista de Buckets muestra las métricas del recuento de errores para la cantidad de buckets que especificó. Puede usar esta información para identificar buckets específicos que están experimentando muchos errores y solucionar los errores por bucket.

Paso 3: solucionar los errores

Tras identificar los buckets con una alta proporción de códigos de estado HTTP específicos, podrá solucionar estos errores. Para obtener más información, consulte los siguientes enlaces:

- ¿Por qué aparece el error 403 Forbidden (Prohibido) cuando intento cargar archivos en Amazon S3?
- ¿Por qué aparece el error 403 Forbidden (Prohibido) cuando intento modificar una política de bucket en Amazon S3?
- ¿Cómo puedo solucionar los errores 403 Forbidden (Prohibido) de mi bucket de Amazon S3, donde todos los recursos provienen de la misma Cuenta de AWS?
- ¿Cómo soluciono un error HTTP 500 o 503 de Amazon S3?

Glosario de métricas de Amazon S3 Storage Lens

El glosario de métricas de Lente de almacenamiento de Amazon S3 proporciona una lista completa de métricas gratuitas y avanzadas de Lente de almacenamiento de S3.

Lente de almacenamiento de S3 ofrece métricas gratuitas para todos los paneles y configuraciones con la opción de actualizar a métricas avanzadas.

- Las métricas gratuitas contienen métricas que son relevantes para su uso de almacenamiento, como la cantidad de buckets y los objetos de la cuenta. Las métricas gratuitas también incluyen métricas basadas en casos de uso, como métricas de optimización de costes y protección de datos. Todas las métricas gratuitas se recopilan a diario y los datos están disponibles para consultas durante 14 días.
- Las métricas y recomendaciones avanzadas incluyen todas las métricas de las métricas gratuitas junto con métricas adicionales, como las métricas de protección de datos y optimización de

costes avanzadas. Las métricas avanzadas también incluyen categorías de métricas adicionales, como métricas de actividad y métricas detalladas de códigos de estado. Los datos de métricas avanzadas están disponibles para consultas durante 15 meses.

Existen cargos adicionales cuando usa S3 Storage Lens con métricas y recomendaciones avanzadas. Para obtener más información, consulte precios de Amazon S3. Para obtener más información sobre características de métricas y recomendaciones avanzadas, consulte Selección de métricas.

> **Note**
>
> Para los grupos de Storage Lens, solo están disponibles las métricas de almacenamiento de nivel gratuito. Las métricas de nivel avanzado no están disponibles en el nivel de grupo de Storage Lens.

Nombres de métricas

La columna Metric name (Nombre de la métrica) de la siguiente tabla proporciona el nombre de cada Lente de almacenamiento de S3 en la consola de S3. La columna de CloudWatch y exportación proporciona el nombre de cada métrica en Amazon CloudWatch y el archivo de exportación de métricas que puede configurar en el panel de Lente de almacenamiento de S3.

Fórmulas de métricas derivadas

Las métricas derivadas no están disponibles para la opción de exportación de métricas y publicación de CloudWatch. Sin embargo, puede utilizar las fórmulas de métricas que se muestran en la columna Derived metrics formula (Fórmula de métricas derivadas) para calcularlas.

Interpretación de los símbolos de prefijo de la Lente de almacenamiento de Amazon S3 para los múltiplos de unidades métricas (K, M, G, etc.)

Los múltiplos de unidades de métricas de Lente de almacenamiento de S3 se escriben con símbolos de prefijo. Estos símbolos de prefijo coinciden con los símbolos del Sistema Internacional de Unidades (SI) que están estandarizados por la Oficina Internacional de Pesos y Medidas (BIPM). También se utilizan estos símbolos en el Código Unificado para Unidades de Medida (UCUM). Para obtener más información, consulte Lista de símbolos de prefijos SI.

> **Note**
>
> - La unidad de medida de los bytes de almacenamiento de S3 está en gigabytes binarios (GB), donde 1 GB equivale a 2^{30} bytes, 1 TB a 2^{40} bytes y 1 PB a 2^{50} bytes. Esta unidad de medida también se conoce como gibibyte (GiB), según la definición de la Comisión Electrotécnica Internacional (IEC).
> - Cuando un objeto llega al final de su vida útil según su configuración de ciclo de vida, Amazon S3 lo coloca en una cola para eliminarlo de manera asincrónica. Por ello, es posible que haya un desfase entre la fecha de vencimiento y la fecha en que Amazon S3 elimina un objeto. S3 Storage Lens no incluye métricas de los objetos que han caducado, pero que no se han eliminado. Para obtener más información sobre las acciones de vencimiento en S3 Lifecycle, consulte Vencimiento de objetos.

Glosario de métricas de Lente de almacenamiento de S3

Nombre de métrica	CloudWatch y exportación	Descripción	Nivel [1]	Cate [2]	D	Fórm métri deriv
Almacenamiento total	StorageBytes	El almacenamiento total, incluidas las cargas incompletas de varias partes, los metadatos de objetos y los marcadores de eliminación	Free	Resu	N	-
Recuento de objetos	ObjectCount	El recuento total de objetos	Free	Resu	N	-
Tamaño de objeto promedio	-	El tamaño de objeto promedio	Free	Resu	Y	sum(StorageBytes)/

Nombre de métrica	CloudWatch y exportación	Descripción	Nivel [1]	Cate [2]	D	Fórm métri deriv
						sum(Objec tCount)
Buckets activos	-	El número total de buckets en uso activo con almacenamiento mayor que 0 bytes	Free	Resu	Y	-
Buckets	-	El número total de buckets	Free	Resu	Y	-
Cuentas	-	El número de cuentas cuyo almacenamiento está en el alcance	Free	Resu	Y	-
Bytes de versión actual	CurrentVe rsionStor ageBytes	El número de bytes que son una versión actual de un objeto	Free	Opti ión de cost	N	-
Porcentaje de bytes de versión actual	-	El porcentaje de bytes en el alcance que son las versiones actuales de objetos	Free	Opti ión de cost	Y	sum(Curre ntVersion StorageBy tes)/ sum(StorageBy tes)
Recuento de objetos de versión actual	CurrentVe rsionObje ctCount	El recuento de los objetos de la versión actual	Free	Opti ión de cost	N	-

Nombre de métrica	CloudWatch y exportación	Descripción	Nivel [1]	Cate [2]	D	Fórm métri deriv
Porcentaje de objetos de versión actual	-	El porcentaje de objetos en el alcance que son una versión actual	Free	Optii ión de cost	Y	sum(Curre ntVersion ObjectCou nt)/ sum(O bjectCoun t)
Bytes de la versión que no es actual	NonCurren tVersionS torageBytes	El número de bytes de la versión que no es actual	Free	Optii ión de cost	N	-
Porcentaje de bytes de la versión que no es actual	-	El porcentaje de bytes en el alcance que son versiones no actuales	Free	Optii ión de cost	Y	sum(NonCu rrentVers ionStorag eBytes)/ s um(Storag eBytes)
Recuento de objetos de versión no actual	NonCurren tVersionO bjectCount	El recuento de las versiones de objetos no actuales	Free	Optii ión de cost	N	-
Porcentaje de objetos de la versión que no es actual	-	El porcentaje de objetos en el alcance que no son una versión actual	Free	Optii ión de cost	Y	sum(NonCu rrentVers ionObject Count)/ su m(ObjectC ount)

Nombre de métrica	CloudWatch y exportación	Descripción	Nivel [1]	Cate [2]	D	Fórm métri deriv
Bytes de marcadores de eliminación	DeleteMarkerStorageBytes	El número de bytes en el alcance que son marcadores de eliminación	Free	Optimización de costo	N	-
Porcentaje de bytes de marcadores de eliminación	-	El porcentaje de bytes en el alcance que son marcadores de eliminación	Free	Optimización de costo	Y	sum(DeleteMarkerStorageBytes)/ sum(StorageBytes)
Recuento de objetos de marcador de eliminación	DeleteMarkerObjectCount	El número total de objetos con un marcador de eliminación	Free	Optimización de costo	N	-
Porcentaje de objetos de marcador de eliminación	-	El porcentaje de objetos en el alcance con un marcador de eliminación	Free	Optimización de costo	Y	sum(DeleteMarkerObjectCount)/ sum(ObjectCount)
Bytes de carga multiparte incompletos	IncompleteMultipartUploadStorageBytes	El total de bytes en el alcance para cargas multiparte incompletas	Free	Optimización de costo	N	-

Nombre de métrica	CloudWatch y exportación	Descripción	Nivel [1]	Cate [2]	D	Fórm métri deriv
Porcentaje de bytes de carga multiparte incompletos	-	El porcentaje de bytes en el alcance que son el resultado de cargas multiparte incompletas	Free	Optii ión de cost	Y	sum(Incom pleteMult ipartUplo adStorage Bytes)/ su m(Storage Bytes)
Recuento de objetos con carga multiparte incompleta	IncompleteMultipartUploadObjectCount	El número de objetos en el alcance que son cargas multiparte incompletas	Free	Optii ión de cost	N	-
Porcentaje de objetos de carga multiparte incompletos	-	El porcentaje de objetos en el alcance que son cargas multiparte incompletas	Free	Optii ión de cost	Y	sum(Incom pleteMult ipartUplo adObjectC ount)/ sum (ObjectCo unt)
Bytes de almacenamiento de carga multiparte incompletos con más de 7 días de antigüedad	IncompleteMPUStorageBytesOlderThan7Days	El total de bytes en alcance para las cargas multiparte incompletas con más de 7 días de antigüedad	Free	Optii ión de cost	N	-

Nombre de métrica	CloudWatch y exportación	Descripción	Nivel [1]	Cate [2]	D	Fórm métri deriv
Porcentaje de bytes de almacenamiento de carga multiparte incompletos con más de 7 días de antigüedad	-	El porcentaje de bytes para cargas multiparte incompletas con más de 7 días de antigüedad	Free	Optimización de costo	Y	sum(IncompleteMPUStorageBytesOlderThan7Days)/sum(StorageBytes)
Recuento de objetos de carga multiparte incompleta con más de 7 días de antigüedad	IncompleteMPUObjectCountOlderThan7Days	El número de objetos que son cargas multiparte incompletas con más de 7 días de antigüedad	Free	Optimización de costo	N	-
Porcentaje del recuento de objetos de carga multiparte incompleta con más de 7 días de antigüedad	-	El porcentaje de objetos que son cargas multiparte incompletas con más de 7 días de antigüedad	Free	Optimización de costo	Y	sum(IncompleteMPUObjectCountOlderThan7Days)/sum(ObjectCount)

Nombre de métrica	CloudWatch y exportación	Descripción	Nivel [1]	Cate [2]	D	Fórm métri deriv
Recuento de reglas del ciclo de vida de la transición	TransitionLifecycleRuleCount	El recuento de reglas del ciclo de vida para la transición de objetos a otra clase de almacenamiento	Adva (Ava)	Opti ión de cost	N	-
Promedio de reglas de ciclo de vida de transición por bucket	-	El número medio de reglas del ciclo de vida para la transición de objetos a otra clase de almacenamiento	Adva (Ava)	Opti ión de cost	Y	sum(TransitionLifecycleRuleCount)/ sum(DistinctNumberOfBuckets)
Recuento de reglas del ciclo de vida del vencimiento	ExpirationLifecycleRuleCount	El recuento de reglas del ciclo de vida para el vencimiento de los objetos	Adva (Ava)	Opti ión de cost	N	-
Promedio de reglas de ciclo de vida de vencimiento por bucket	-	El número medio de reglas del ciclo de vida para el vencimiento de los objetos	Adva (Ava)	Opti ión de cost	Y	sum(ExpirationLifecycleRuleCount)/ sum(DistinctNumberOfBuckets)

Nombre de métrica	CloudWatch y exportación	Descripción	Nivel [1]	Cate [2]	D	Fórm métri deriv
Recuento de reglas de ciclo de vida de transición de versiones no actuales	NoncurrentVersionTransitionLifecycleRuleCount	El recuento de reglas del ciclo de vida para la transición de versiones de objetos no actuales a otra clase de almacenamiento	Adva (Ava)	Opti ión de cost	N	
Promedio de reglas del ciclo de vida de transición de versión no actual por bucket	-	El número medio de reglas del ciclo de vida para la transición de versiones de objetos no actuales a otra clase de almacenamiento	Adva (Ava)	Opti ión de cost	Y	sum(NoncurrentVersionTransitionLifecycleRuleCount)/sum(DistinctNumberOfBuckets)
Recuento de reglas del ciclo de vida de vencimiento de versiones no actuales	NoncurrentVersionExpirationLifecycleRuleCount	El recuento de reglas del ciclo de vida para el vencimiento de versiones de objetos no actuales	Adva (Ava)	Opti ión de cost	N	-

Nombre de métrica	CloudWatch y exportación	Descripción	Nivel [1]	Cate [2]	D	Fórm métri deriv
Promedio de reglas del ciclo de vida de vencimiento de versión no actual por bucket	-	El número medio de reglas del ciclo de vida para el vencimiento de versiones de objetos no actuales	Adva (Ava)	Opti ión de cost	Y	sum(Noncu rrentVers ionExpira tionLifec ycleRuleC ount)/ sum (Distinct NumberOfB uckets)
Abortar el recuento de reglas del ciclo de vida de carga multiparte incompleta	AbortInco mpleteMPU Lifecycle RuleCount	El recuento de reglas del ciclo de vida para eliminar cargas multiparte incompletas	Adva (Ava)	Opti ión de cost	N	-
Promedio de reglas del ciclo de vida de carga multiparte incompleta de anulación por bucket	-	El número medio de reglas del ciclo de vida para eliminar cargas multiparte incompletas	Adva (Ava)	Opti ión de cost	Y	sum(Abort Incomplet eMPULifec ycleRuleC ount)/ sum (Distinct NumberOfB uckets)

Nombre de métrica	CloudWatch y exportación	Descripción	Nivel [1]	Cate [2]	D	Fórm métri deriv
Recuento de reglas del ciclo de vida del marcador de eliminación de objetos vencidos	ExpiredObjectDeleteMarkerLifecycleRuleCount	El recuento de reglas del ciclo de vida para eliminar marcadores de eliminación de objetos vencidos	Adva (Ava)	Opti ión de cost	N	-
Promedio de reglas de ciclo de vida de marcadores de eliminación de objetos vencidos por bucket	-	El número medio de reglas del ciclo de vida para eliminar marcadores de eliminación de objetos vencidos	Adva (Ava)	Opti ión de cost	Y	sum(ExpiredObjectDeleteMarkerLifecycleRuleCount)/sum(DistinctNumberOfBuckets)
Recuento de reglas del ciclo de vida total	TotalLifecycleRuleCount	El recuento total de las reglas del ciclo de vida	Adva (Ava)	Opti ión de cost	N	-
Promedio de recuento de reglas del ciclo de vida por bucket	-	El número medio de reglas del ciclo de vida	Adva (Ava)	Opti ión de cost	Y	sum(TotalLifecycleRuleCount)/sum(DistinctNumberOfBuckets)

Nombre de métrica	CloudWatch y exportación	Descripción	Nivel [1]	Cate [2]	D	Fórm métri deriv
Bytes cifrados	EncryptedStorageBytes	El número total de bytes cifrados	Free	Prot n de los dato	N	-
Porcentaje de bytes cifrados	-	El porcentaje total de bytes que están cifrados	Free	Prot n de los dato	Y	sum(EncryptedObjectCount)/sum(StorageBytes)
Recuento de objetos cifrados	EncryptedObjectCount	El recuento total de los objetos que están cifrados	Free	Prot n de los dato	N	-
Porcentaje de objetos cifrados	-	El porcentaje de objetos que están cifrados	Free	Prot n de los dato	Y	sum(EncryptedStorageBytes)/sum(ObjectCount)
Bytes sin cifrar	UnencryptedStorageBytes	El número de bytes que no están cifrados	Free	Prot n de los dato	Y	sum(StorageBytes) - sum(EncryptedStorageBytes)

Nombre de métrica	CloudWatch y exportación	Descripción	Nivel [1]	Cate [2]	D	Fórm métri deriv
Porcentaje de bytes sin cifrar	-	El porcentaje de bytes en el alcance que no están cifrados	Free	Prot n de los dato	Y	sum(Unenc ryptedSto rageBytes)/ sum(Sto rageBytes)
Recuento de objetos sin cifrar	UnencryptedObjectCount	El recuento total de objetos que no están cifrados	Free	Prot n de los dato	Y	sum(Objec tCount) - sum(Encry ptedObjec tCount)
Porcentaje de objetos sin cifrar	-	El porcentaje de objetos sin cifrar	Free	Prot n de los dato	Y	sum(Unenc ryptedSto rageBytes)/ sum(Obj ectCount)
Origen de bytes de almacenamiento replicados	ReplicatedStorageBytesSource	El número total de bytes que se replican desde el bucket de origen	Free	Prot n de los dato	N	-

Nombre de métrica	CloudWatch y exportación	Descripción	Nivel [1]	Cate [2]	D	Fórm métri deriv
Porcentaje de bytes replicados de origen	-	El porcentaje total de bytes que se replican desde el bucket de origen	Free	Prot n de los dato	Y	sum(ReplicatedStorageBytesSource)/ sum(StorageBytes)
Origen del recuento de objetos replicados	ReplicatedObjectCountSource	El recuento de objetos replicados del bucket de origen	Free	Prot n de los dato	N	-
Porcentaje de objetos replicados de origen	-	El porcentaje total de objetos que se replican desde el bucket de origen	Free	Prot n de los dato	Y	sum(ReplicatedStorageObjectCount)/ sum(ObjectCount)
Replicación de bytes de almacenamiento en destino	ReplicatedStorageBytes	El número total de bytes que se replican en el bucket de destino	Free	Prot n de los dato	Y	-

Nombre de métrica	CloudWatch y exportación	Descripción	Nivel [1]	Cate [2]	D	Fórm métri deriv
Porcentaje de bytes replicados en destino	-	El porcentaje de bytes totales que se replican en el bucket de destino	Free	Prot n de los dato	Y	sum(Repli catedStor ageBytes) / sum(Stor ageBytes)
Recuento de objetos replicados en destino	ReplicatedObjectCount	El recuento de objetos que se replican en el bucket de destino	Free	Prot n de los dato	Y	-
Porcentaje de objetos replicados en destino	-	El porcentaje de objetos totales que se replican en el bucket de destino	Free	Prot n de los dato	Y	sum(Repli catedObje ctCount)/ sum(Objec tCount)
Bytes de Object Lock	ObjectLockEnabledStorageBytes	El recuento total de bytes de almacenamiento habilitados para Object Lock	Free	Prot n de los dato	Y	sum(Unenc ryptedSto rageBytes)/ sum(Obj ectLockEn abledStor ageCount) - sum(Obje ctLockEna bledStora geBytes)

Nombre de métrica	CloudWatch y exportación	Descripción	Nivel [1]	Cate [2]	D	Fórm métri deriv
Porcentaje de bytes de Object Lock	-	El porcentaje de bytes de almacenamiento habilitados para Object Lock	Free	Prot n de los dato	Y	sum(ObjectLockEnabledStorageBytes)/sum(StorageBytes)
Recuento de objetos de Object Lock	ObjectLockEnabledObjectCount	El recuento total de objetos de Object Lock	Free	Prot n de los dato	Y	-
Porcentaje de objetos de Object Lock	-	El porcentaje de objetos totales que tienen habilitado Object Lock	Free	Prot n de los dato	Y	sum(ObjectLockEnabledObjectCount)/sum(ObjectCount)
Recuento de buckets con control de versiones habilitado	VersioningEnabledBucketCount	El recuento de buckets que tienen habilitado el control de versiones de S3	Free	Prot n de los dato	N	-

Nombre de métrica	CloudWatch y exportación	Descripción	Nivel [1]	Cate [2]	D	Fórm métri deriv
Porcentaje de buckets con control de versiones habilitado	-	El porcentaje de buckets que tienen habilitado el control de versiones de S3	Free	Prot n de los dato	Y	sum(VersioningEnabledBucketCount)/ sum(DistinctNumberOfBuckets)
Recuento de buckets que tienen habilitado la eliminación de MFA	MFADeleteEnabledBucketCount	El recuento de buckets que tienen habilitado la eliminación de MFA (autenticación multifactor)	Free	Prot n de los dato	N	-
Porcentaje de buckets que tienen habilitado la eliminación de MFA	-	El porcentaje de buckets que tienen habilitado la eliminación de MFA (autenticación multifactor)	Free	Prot n de los dato	Y	sum(MFADeleteEnabledBucketCount)/ sum(DistinctNumberOfBuckets)

Nombre de métrica	CloudWatch y exportación	Descripción	Nivel [1]	Cate [2]	D	Fórm métri deriv
Recuento de buckets con SSE-KMS habilitado	SSEKMSEnabledBucketCount	El recuento de buckets que utilizan el cifrado del lado del servidor con claves de AWS Key Management Service (SSE-KMS) para el cifrado de buckets predeterminado	Free	Prot n de los dato	N	-
Porcentaje de buckets con SSE-KMS habilitado	-	El porcentaje de buckets con SSE-KMS para el cifrado de buckets predeterminado	Free	Prot n de los dato	Y	sum(SSEKMSEnabledBucketCount)/sum(DistinctNumberOfBuckets)
Todas las solicitudes de firma no admitidas	AllUnsupportedSignatureRequests	El número total de solicitudes que utilizan versiones de firma de AWS no compatibles	Adva (Ava)	Prot n de los dato	N	-

Nombre de métrica	CloudWatch y exportación	Descripción	Nivel [1]	Cate [2]	D	Fórm métri deriv
Porcentaje de todas las solicitudes de firma no admitidas	-	El porcentaje de solicitudes que utilizan versiones de firma de AWS no admitidas	Adva (Ava)	Prot n de los dato	Y	sum(AllUn supported SignatureRequests) / sum(AllR equests)
Todas las solicitudes de TLS no admitidas	AllUnsupp ortedTLSR equests	La cantidad de solicitudes que utilizan versiones de seguridad de la capa de transporte (TLS) no admitidas	Adva (Ava)	Prot n de los dato	N	-
Porcentaje de todas las solicitudes de TLS no admitidas	-	El porcentaj e de solicitud es que utilizan versiones de TLS no admitidas	Adva (Ava)	Prot n de los dato	Y	sum(AllUn supported TLSReques ts)/ sum(A llRequest s)
Todas las solicitudes de SSE-KMS	AllSSEKMS Requests	El número total de solicitudes que especifican SSE-KMS	Adva (Ava)	Prot n de los dato	N	-

Nombre de métrica	CloudWatch y exportación	Descripción	Nivel [1]	Cate [2]	D	Fórm métri deriv
Porcentaje de todas las solicitudes de SSE-KMS	-	El porcentaje de solicitudes que especifican SSE-KMS	Adva (Ava)	Prot n de los dato	Y	sum(AllSSEKMSRequests)/sum(AllRequests)
Recuento de reglas de replicación de la misma región	SameRegionReplicationRuleCount	El recuento de reglas de replicación para la replicación de la misma región (SRR)	Adva (Ava)	Prot n de los dato	N	-
Promedio de reglas de replicación de la misma región por bucket	-	El número medio de reglas de replicación para SRR	Adva (Ava)	Prot n de los dato	Y	sum(SameRegionReplicationRuleCount)/sum(DistinctNumberOfBuckets)
Recuento de reglas de replicación entre regiones	CrossRegionReplicationRuleCount	El recuento de reglas de replicación para la replicación entre regiones (CRR)	Adva (Ava)	Prot n de los dato	N	-

Nombre de métrica	CloudWatch y exportación	Descripción	Nivel [1]	Cate [2]	D	Fórm métri deriv
Promedio de reglas de replicación entre regiones por bucket	-	El número medio de reglas de replicación para CRR	Adva (Ava)	Prot n de los dato	Y	sum(Cross RegionRep licationR uleCount) / sum(Dist inctNumbe rOfBucket s)
Recuento de reglas de replicación de la misma cuenta	SameAccou ntReplica tionRuleCount	El recuento de reglas de replicación para la replicación dentro de la misma cuenta	Adva (Ava)	Prot n de los dato	N	-
Promedio de reglas de replicación de la misma cuenta por bucket	-	El número medio de reglas de replicación para la replicación dentro de la misma cuenta	Adva (Ava)	Prot n de los dato	Y	sum(SameA ccountRep licationR uleCount) / sum(Dist inctNumbe rOfBucket s)
Recuento de reglas de replicación entre cuentas	CrossAcco untReplic ationRuleCount	El recuento de reglas de replicación para la replicación entre cuentas	Adva (Ava)	Prot n de los dato	N	-

Nombre de métrica	CloudWatch y exportación	Descripción	Nivel [1]	Cate [2]	D	Fórm métri deriv
Promedio de reglas de replicación entre cuentas por bucket	-	El número medio de reglas de replicación para la replicación entre cuentas	Adva (Ava)	Prot n de los dato	Y	sum(Cross AccountRe plication RuleCount)/ sum(Dis tinctNumb erOfBucke ts)
Recuento de reglas de replicación de destino no válido	InvalidDe stination Replicati onRuleCount	El recuento de reglas de replicaci ón con un destino de replicación que no es válido	Adva (Ava)	Prot n de los dato	N	-
Promedio de reglas de replicación de destinos no válidos por bucket	-	El número medio de reglas de replicación con un destino de replicación que no es válido	Adva (Ava)	Prot n de los dato	Y	sum(Inval idReplica tionRuleC ount)/ sum (Distinct NumberOfB uckets)
Recuento de reglas de replicación total	-	El recuento de reglas de replicaci ón total	Adva (Ava)	Prot n de los dato	Y	-

Nombre de métrica	CloudWatch y exportación	Descripción	Nivel [1]	Cate [2]	D	Fórm métri deriv
Promedio de recuento de reglas de replicación por bucket	-	El promedio de recuento de reglas de replicación total	Adva (Ava)	Prot n de los dato	Y	sum(all replicati on rule count metrics)/ sum(Disti nctNumber OfBuckets)
Recuento de buckets de la propiedad de objetos del propietario del bucket obligator io	ObjectOwn ershipBuc ketOwnerE nforcedBu cketCount	El recuento total de buckets que tienen listas de control de acceso (ACL) desactiva das al usar la configuración del propietario del bucket obligatorio para la propiedad de objetos	Free	Adm ació de acce	N	-
Porcentaje de buckets del propietar io del bucket obligatorio de la propiedad de objetos	-	El porcentaj e de buckets que tienen ACL desactivadas al usar la configura ción de propietar io del bucket obligatorio para la propiedad de objetos	Free	Adm ació de acce	Y	sum(Objec tOwnershi pBucketOw nerEnforc edBucketC ount)/ sum (Distinct NumberOfB uckets)

Nombre de métrica	CloudWatch y exportación	Descripción	Nivel [1]	Cate [2]	D	Fórm métri deriv
Recuento de buckets de la propiedad de objetos del propietario del bucket preferido	ObjectOwnershipBucketOwnerPreferredBucketCount	El recuento total de buckets que utilizan la configuración de propietario del bucket preferido para la propiedad de objetos	Free	Adm ació de acce	N	-
Porcentaje de buckets de la propiedad de objetos del propietario del bucket preferido	-	El porcentaje de buckets que utilizan la configuración de propietario del bucket preferido para la propiedad de objetos	Free	Adm ació de acce	Y	sum(ObjectOwnershipBucketOwnerPreferredBucketCount)/ sum(DistinctNumberOfBuckets)
Recuento de buckets del escritor de objetos de la propiedad de objetos	ObjectOwnershipObjectWriterBucketCount	El recuento total de buckets que utilizan la configuración del escritor de objetos para la propiedad de objetos	Free	Adm ació de acce	N	-

Nombre de métrica	CloudWatch y exportación	Descripción	Nivel [1]	Cate [2]	D	Fórm métri deriv
Porcentaje de buckets del escritor de objetos de la propiedad de objetos	-	El porcentaj e de buckets que utilizan la configuración del escritor de objetos para la propiedad de objetos	Free	Adm ació de acce	Y	sum(Objec tOwnershi pObjectWr iterBucke tCount)/ s um(Distin ctNumberO fBuckets)
Recuento de buckets con Transfer Acceleration habilitada	TransferA ccelerati onEnabled BucketCount	El recuento total de buckets que tienen Transfer Acceleration habilitada	Free	Dese	N	-
Porcentaje de buckets con Transfer Acceleration habilitada	-	El porcentaje de buckets que tienen Transfer Acceleration habilitada	Free	Dese	Y	sum(Trans ferAccele rationEna bledBucke tCount)/ s um(Distin ctNumberO fBuckets)
Recuento de bucket con la notificación de eventos habilitada	EventNoti ficationE nabledBuc ketCount	El recuento total de buckets que tienen habilitadas las notificaciones de eventos	Free	Ever	N	

Nombre de métrica	CloudWatch y exportación	Descripción	Nivel [1]	Cate [2]	D	Fórm métri deriv
Porcentaje de buckets con la notificación de eventos habilitada	-	El porcentaje de buckets que tienen habilitadas las notificaciones de eventos	Free	Ever	Y	sum(Event NotificationEnabledBucketCount)/sum(DistinctNumberOfBuckets)
Todas las solicitudes	AllRequests	El número total de solicitudes realizadas	Adva (Ava)	Activ	N	-
Solicitudes GET	GetRequests	El número total de solicitudes `GET` realizadas	Adva (Ava)	Activ	N	-
Solicitudes PUT	PutRequests	El número total de solicitudes `PUT` realizadas	Adva (Ava)	Activ	N	-
Solicitudes HEAD	HeadRequests	El número total de solicitudes `HEAD` realizadas	Adva (Ava)	Activ	N	-
Solicitudes DELETE	DeleteRequests	El número total de solicitudes `DELETE` realizadas	Adva (Ava)	Activ	N	-
Solicitudes LIST	ListRequests	El número total de solicitudes `LIST` realizadas	Adva (Ava)	Activ	N	-

Nombre de métrica	CloudWatch y exportación	Descripción	Nivel [1]	Cate [2]	D	Fórm métri deriv
Solicitudes POST	PostRequests	El número total de solicitudes POST realizadas	Adva (Ava)	Activ	N	-
Solicitudes SELECT	SelectRequests	El número total de solicitudes SELECT de S3	Adva (Ava)	Activ	N	-
Seleccion ar bytes escaneados	SelectSca nnedBytes	El número de bytes seleccion ados de S3 analizados	Adva (Ava)	Activ	N	-
Seleccionar bytes devueltos	SelectRet urnedBytes	El número de bytes seleccion ados de S3 devueltos	Adva (Ava)	Activ	N	-
Bytes descargados	BytesDown loaded	El número de bytes descargad os	Adva (Ava)	Activ	N	-
Porcentaje de tasa de recuperación	-	El porcentaje de bytes descargad os	Adva (Ava)	Activ	Y	sum(Bytes Downloade d)/ sum(St orageByte s)
Bytes cargados	BytesUploaded	El número de bytes cargados	Adva (Ava)	Activ	N	-

Nombre de métrica	CloudWatch y exportación	Descripción	Nivel [1]	Cate [2]	D	Fórm métri deriv
Porcentaje de tasa de incorporación	-	El porcentaje de bytes cargados	Adva (Ava)	Activ	Y	sum(Bytes Uploaded) / sum(Stor ageBytes)
Errores 4xx	4xxErrors	El número total de códigos de estado HTTP 4xx	Adva (Ava)	Activ	N	-
Errores 5xx	5xxErrors	El número total de códigos de estado HTTP 5xx	Adva (Ava)	Activ	N	-
Errores totales	-	La suma de todos los errores 4xx y 5xx	Adva (Ava)	Activ	Y	sum(4xxEr rors) + sum(5xxEr rors)
Porcentaje de tasa de errores	-	El número total de errores 4xx y 5xx como porcentaj e del total de solicitudes	Adva (Ava)	Activ	Y	sum(Total Errors)/ s um(TotalR equests)
Recuento de estado 200 OK	200OKStat usCount	El recuento total de códigos de estado 200 OK	Adva (Ava)	Códi de esta deta	N	-

Nombre de métrica	CloudWatch y exportación	Descripción	Nivel [1]	Cate [2]	D	Fórm métri deriv
Porcentaje de estado 200 OK	-	El número total de códigos de estado 200 OK como porcentaje del total de solicitudes	Adva (Ava)	Códi de esta deta	Y	sum(200OK StatusCou nt)/ sum(A llRequest s)
Código de estado 206 Partial Content (Contenido parcial)	206Partia lContentS tatusCount	El recuento total de códigos de estado 206 Partial Content (Contenid o parcial)	Adva (Ava)	Códi de esta deta	N	-
Porcentaje de estado de contenido parcial	-	El número total de códigos de estado 206 Contenido parcial como porcentaje del total de solicitudes	Adva (Ava)	Códi de esta deta	Y	sum(206Pa rtialCont entStatus Count)/ su m(AllRequ ests)
Recuento de error 400 Bad Request (Solicitud errónea)	400BadReq uestErrorCount	El recuento total de códigos de estado 400 Solicitud errónea	Adva (Ava)	Códi de esta deta	N	-
Porcentaje de errores 400 Bad Request (Solicitud errónea)	-	El número total de códigos de estado 400 Solicitud errónea como porcentaje del total de solicitudes	Adva (Ava)	Códi de esta deta	Y	sum(400Ba dRequestE rrorCount)/ sum(All Requests)

Nombre de métrica	CloudWatch y exportación	Descripción	Nivel [1]	Cate [2]	D	Fórm métri deriv
Recuento de errores 403 Forbidden (Prohibido)	403ForbiddenErrorCount	El recuento total de códigos de estado 403 Forbidden (Prohibido)	Adva (Ava)	Códi de esta deta	N	-
Porcentaje de errores 403 Forbidden (Prohibido)	-	El número total de códigos de estado 403 Forbidden (Prohibido) como porcentaje del total de solicitudes	Adva (Ava)	Códi de esta deta	Y	sum(403ForbiddenErrorCount) / sum(AllRequests)
Recuento de errores 404 Not Found (No encontrado)	404NotFoundErrorCount	El recuento total de códigos de estado 404 Not Found (No encontrado)	Adva (Ava)	Códi de esta deta	N	-
Porcentaje de errores 404 Not Found (No encontrado)	-	El número total de códigos de estado 404 Not Found (No encontrado) como porcentaje del total de solicitudes	Adva (Ava)	Códi de esta deta	Y	sum(404NotFoundErrorCount)/ sum(AllRequests)
Recuento de error 500 Internal Server (Servidor interno)	500InternalServerErrorCount	El recuento total de códigos de estado 500 Internal Server Error (Error de servidor interno)	Adva (Ava)	Códi de esta deta	N	-

Nombre de métrica	CloudWatch y exportación	Descripción	Nivel [1]	Cate [2]	D	Fórm métri deriv
Porcentaje de errores 500 Internal Server (Servidor interno)	-	El número total de códigos de estado 500 Internal Server Error (Error de servidor interno) como porcentaje del total de solicitudes	Adva (Ava)	Códi de esta deta	Y	sum(500In ternalSer verErrorC ount)/ sum (AllReque sts)
Recuento de errores 503 Service Unavailable (Servicio no disponible)	503Servic eUnavaila bleErrorCount	El recuento total de códigos de estado 503 Service Unavailab le (Servicio no disponible)	Adva (Ava)	Códi de esta deta	N	-
Porcentaje de errores 503 Service Unavailable (Servicio no disponible)	-	El número total de códigos de estado 503 Service Unavailab le (Servicio no disponible) como porcentaje del total de solicitudes	Adva (Ava)	Códi de esta deta	Y	sum(503Se rviceUnav ailableEr rorCount) / sum(AllR equests)

[1] Todas las métricas de almacenamiento de nivel gratuito están disponibles en el nivel de grupo de Storage Lens. Las métricas de nivel avanzado no están disponibles en el nivel de grupo de Storage Lens.

[2] Las métricas de recuento de reglas y las métricas de configuración de buckets no están disponibles en el nivel de prefijo.

Trabajo con Amazon S3 Storage Lens mediante la consola y la API

La Lente de almacenamiento de Amazon S3 es una función de análisis de almacenamiento en la nube que puede utilizar para obtener visibilidad en toda la organización sobre el uso y la actividad del almacenamiento de objetos. Puede utilizar las métricas de Lente de almacenamiento de S3 para generar información resumida, como averiguar cuánto almacenamiento tiene en toda la organización o cuáles son los buckets y los prefijos de crecimiento más rápido. También puede utilizar las métricas de Lente de almacenamiento de S3 para identificar oportunidades de optimización de costos, implementar las prácticas recomendadas de protección y seguridad de los datos y mejorar el rendimiento de las cargas de trabajo de las aplicaciones. Por ejemplo, puede identificar los buckets que no tienen reglas del ciclo de vida de S3 para que hagan vencer las cargas multipartes incompletas que tengan más de 7 días de antigüedad. También puede identificar los buckets que no siguen las prácticas recomendadas de protección de datos, como el uso de la Replicación de S3 o el control de versiones de S3. Lente de almacenamiento de S3 analiza también las métricas para ofrecer recomendaciones contextuales que puede usar para optimizar los costos de almacenamiento y aplicar las prácticas recomendadas para proteger los datos.

Lente de almacenamiento de S3 agrega las métricas y muestra la información en la sección Account snapshot (Instantánea de la cuenta) en la página Buckets de la consola de Amazon S3. Lente de almacenamiento de S3 también proporciona un panel interactivo que puede utilizar para visualizar información y tendencias, marcar valores atípicos y recibir recomendaciones para optimizar los costos de almacenamiento y aplicar las prácticas recomendadas de protección de datos. El panel tiene opciones de desglose para generar y visualizar información en el nivel de la organización, la cuenta, la Región de AWS, la clase de almacenamiento, el bucket, el prefijo o el grupo de Storage Lens. También puede enviar una exportación de métricas diaria en formato CSV o Parquet a un bucket de S3.

Las siguientes secciones contienen ejemplos de la creación, la actualización y la visualización de las configuraciones de S3 Storage Lens y de la realización de operaciones relacionadas con la característica. Si utiliza S3 Storage Lens con AWS Organizations, estos ejemplos también abarcan esos casos de uso. En los ejemplos, reemplace los valores de las variables por aquellos que sean específicos para usted.

Temas

Uso de Amazon S3 Storage Lens en la consola

Amazon S3 Storage Lens es una función de análisis de almacenamiento en la nube que puede utilizar para obtener visibilidad en toda la organización sobre el uso y la actividad del almacenamiento de objetos. Puede utilizar las métricas de S3 Storage Lens para generar información resumida, como averiguar cuánto almacenamiento tiene en toda la organización o cuáles son los buckets y los prefijos de crecimiento más rápido. También puede utilizar las métricas de Amazon S3 Storage Lens para identificar oportunidades de optimización de costos, implementar las prácticas recomendadas de protección y seguridad de los datos y mejorar el rendimiento de las cargas de trabajo de las aplicaciones. Por ejemplo, puede identificar los buckets que no tienen reglas del ciclo de vida de S3 para que hagan vencer las cargas multipartes incompletas que tengan más de 7 días de antigüedad. También puede identificar los buckets que no siguen las prácticas recomendadas de protección de datos, como el uso de la Replicación de S3 o el control de versiones de S3. S3 Storage Lens analiza también las métricas para ofrecer recomendaciones contextuales que puede usar para optimizar los costos de almacenamiento y aplicar las prácticas recomendadas para proteger los datos.

S3 Storage Lens agrega las métricas y muestra la información en la sección Instantánea de la cuenta en la página Buckets de la consola de Amazon S3. S3 Storage Lens también proporciona un panel interactivo que puede utilizar para visualizar información y tendencias, marcar valores atípicos y recibir recomendaciones para optimizar los costes de almacenamiento y aplicar las prácticas recomendadas de protección de datos. El panel tiene opciones de desglose para generar y visualizar información en el nivel de la organización, la cuenta, la Región de AWS, la clase de almacenamiento, el bucket, el prefijo o el grupo de Lente de almacenamiento. También puede enviar una exportación de métricas diaria en formato CSV o Parquet a un bucket de S3.

> **Note**
>
> Las actualizaciones en la configuración del panel pueden tardar hasta 48 horas en mostrarse o visualizarse con precisión.

Temas

Creación y actualización de los paneles de Amazon S3 Storage Lens

Lente de almacenamiento de S3 agrega las métricas y muestra la información en la sección Account snapshot (Instantánea de la cuenta) en la página Buckets de la consola de Amazon S3. S3 Storage Lens también proporciona un panel interactivo que puede utilizar para visualizar información y tendencias, marcar valores atípicos y recibir recomendaciones para optimizar los costes de almacenamiento y aplicar las prácticas recomendadas de protección de datos. El panel tiene opciones de desglose para generar y visualizar información en el nivel de la organización, la cuenta, la Región de AWS, la clase de almacenamiento, el bucket, el prefijo o el grupo de Lente de almacenamiento. También puede enviar una exportación de métricas diaria en formato CSV o Parquet a un bucket de S3.

El panel predeterminado de Amazon S3 Storage Lens es default-account-dashboard. Amazon S3 preconfigura este panel para ayudarle a visualizar información resumida y tendencias para las métricas avanzadas y gratuitas agregadas de toda la cuenta en la consola. No puede modificar el alcance de configuración del panel, pero puede actualizar la selección de métricas de métricas gratuitas a métricas y recomendaciones avanzadas pagadas, configurar la exportación de métricas opcionales o incluso desactivar el panel predeterminado. El panel predeterminado no se puede eliminar.

También puede crear paneles personalizados de Lente de almacenamiento de S3 adicionales que pueden abarcar a la organización en AWS Organizations o a regiones específicas o buckets dentro de una cuenta.

Creación de un panel de Amazon S3 Storage Lens

Siga los pasos que figuran a continuación para crear un panel de Amazon S3 Storage Lens en la consola de Amazon S3.

Paso 1: definir el alcance del panel

1. Inicie sesión AWS Management Console Management Console y abra la consola de Amazon S3 en https://console.aws.amazon.com/s3/.
2. En la barra de navegación de la parte superior de la página, elija el nombre de la región de AWS que aparece. A continuación, elija la región a la que desea cambiar.
3. En el panel de navegación, en Lente de almacenamiento de S3, elija Paneles.
4. Elija Create dashboard (Crear un panel).
5. En la página Dashboard (Panel), en la sección General, haga lo siguiente:

a. Vea la región de origen del panel. La región de origen es la Región de AWS donde se almacenan la configuración y las métricas de este Almacenamiento de lente.

b. Escriba un nombre de panel.

   Los nombres de los paneles deben tener menos de 65 caracteres y no deben contener caracteres ni espacios especiales.

   > **Note**
   >
   > No puede cambiar el nombre de este panel una vez que se creado.

c. Si lo desea, puede agregar etiquetas al panel. Puede usar etiquetas para administrar los permisos de su panel y realizar un seguimiento de los costos de S3 Storage Lens.

   Para obtener más información, consulte Control del acceso mediante etiquetas de recursos en la Guía del usuario de IAM y Etiquetas de asignación de costos generadas por AWS en la Guía del usuario de AWS Billing.

   > **Note**
   >
   > Puede agregar hasta 50 etiquetas a la configuración del panel.

6. En la sección Dashboard scope(Alcance del panel), haga lo siguiente:

   a. Elija las regiones y los buckets que desea que S3 Storage Lens incluya o excluya en el panel.

   b. Elija los buckets en las regiones seleccionadas que desea que S3 Storage Lens incluya o excluya. Puede incluir o excluir buckets, pero no ambos. Esta opción no está disponible cuando se crean paneles de nivel de organización.

> **Note**
>
> - Puede incluir o excluir regiones y buckets. Esta opción está limitada a las regiones solo cuando se crean paneles de nivel de organización en las cuentas de miembros de su organización.
> - Puede elegir hasta 50 buckets para incluir o excluir.

Paso 2: configurar la selección de métricas

1. En la sección Metrics selection (Selección de Métricas), elija el tipo de métricas que desea agregar a este panel.

   - Para incluir métricas gratuitas agregadas en el nivel de bucket y disponibles para consultas durante 14 días, elija Free Metrics (Métricas gratuitas).
   - Para habilitar las métricas avanzadas y otras opciones avanzadas, elija Advanced metrics and recommendations (Métricas y recomendaciones avanzadas). Estas opciones incluyen la agregación de prefijos, la publicación en Amazon CloudWatch y las recomendaciones contextuales avanzadas. Los datos están disponibles para consultas durante 15 meses. Las métricas y recomendaciones avanzadas tienen un costo adicional. Para obtener más información, consulte precios de Amazon S3.

     Para obtener más información sobre las métricas avanzadas y gratuitas, consulte Selección de métricas.
2. En Advanced metrics and recommendations features (Características de métricas y recomendaciones avanzadas), seleccione las opciones que desea habilitar:

   - Advanced metrics (Métricas avanzadas)
   - Publicación de CloudWatch
   - Agregación de prefijos

   > **Important**
   >
   > Si habilita la agregación de prefijos para la configuración de S3 Storage Lens, las métricas de nivel de prefijo no se publicarán en CloudWatch. En CloudWatch, solo se publican métricas de S3 Storage Lens a nivel de bucket, cuenta y organización.
3. Si ha habilitado Advanced metrics (Métricas avanzadas), seleccione las Advanced metrics categories (Categorías de métricas avanzadas) que desea mostrar en el panel de Lente de almacenamiento de S3:

   - Métricas de actividad
   - Detailed status code metrics (Métricas de código de estado detalladas)
   - Advanced cost optimization metrics (Métricas de optimización de costos avanzadas)
   - Advanced data protection metrics (Métricas de protección de datos avanzadas)

Para obtener más información sobre categorías de métricas, consulte Categorías de métricas. Para obtener una lista completa de métricas, consulte Glosario de métricas de Amazon S3 Storage Lens.

4. Si eligió habilitar la agregación de prefijos, configure lo siguiente:

   a. Elija el tamaño mínimo de umbral de prefijo para este panel.

      Por ejemplo, un umbral de prefijo del 5 por ciento indica que se agregarán los prefijos que componen el 5 por ciento o más del tamaño del almacenamiento total del bucket.

   b. Elija la profundidad del prefijo.

      Esta configuración indica el número máximo de niveles hasta los que se evalúan los prefijos. La profundidad del prefijo debe ser inferior a 10.

   c. Escriba un carácter delimitador de prefijo.

      Este valor se utiliza para identificar cada nivel de prefijo. El valor predeterminado de Amazon S3 es el carácter `/`, pero su estructura de almacenamiento podría utilizar otros caracteres delimitadores.

(Opcional) Paso 3: exportar métricas para el panel

1. En la sección Metrics Export (Exportación de métricas), para crear una exportación de métricas que se colocará diariamente en un bucket de destino de su elección, elija Enable (Habilitar).

   La exportación de métricas está en formato CSV o Apache Parquet. Representa el mismo alcance de datos que los datos del panel de S3 Storage Lens sin las recomendaciones.

2. Si ha habilitado la exportación de métricas, elija el formato de salida de la exportación diaria de métricas: CSV o Apache Parquet.

   Parquet es un formato de archivo de código abierto para Hadoop que almacena los datos anidados en un formato de columna plano.

3. Elija el bucket de S3 de destino para la exportación de métricas.

   Puede elegir un bucket en la cuenta actual del panel S3 Storage Lens. También puede elegir otra Cuenta de AWS si tiene los permisos del bucket de destino y el ID de la cuenta del propietario del bucket de destino.

4. Elija el bucket de S3 de destino (formato: `s3://bucket-name/prefix`).

   El bucket debe estar en la región de inicio del panel de Lente de almacenamiento de S3. La consola de S3 le muestra el Destination bucket permission (Permiso del bucket de destino) que agregará Amazon S3 a la política de bucket de destino. Amazon S3 actualiza la política de bucket en el bucket de destino para permitir que S3 coloque datos en ese bucket.

5. (Opcional) Para habilitar el cifrado del lado del servidor para la exportación de métricas, elija Specify an encryption key (Especificar una clave de cifrado). A continuación, elija el tipo de cifrado: Claves administradas de Amazon S3 (SSE-S3) o Clave de AWS Key Management Service (SSE-KMS).

   Puede elegir entre una clave administrada de Amazon S3 (SSE-S3) y una clave de AWS Key Management Service (AWS KMS) (SSE-KMS).

6. (Opcional) Para especificar una clave de AWS KMS, debe elegir una clave KMS o ingresar un nombre de recurso de Amazon (ARN) de clave.

   Si elige una clave administrada por el cliente, debe conceder permiso a Lente de almacenamiento de S3 para cifrar la política de claves de AWS KMS. Para obtener más información, consulte Uso de una AWS KMS key para cifrar las exportaciones de métricas.

7. Elija Create dashboard (Crear un panel).

Para lograr una mayor visibilidad de su almacenamiento, puede crear uno o más grupos de Lente de almacenamiento de S3 y asociarlos a su panel. Un grupo de S3 Lente de almacenamiento es un filtro definido y personalizado para objetos basado en prefijos, sufijos, etiquetas de objetos, tamaño de objetos, antigüedad de objetos o una combinación de estos filtros.

Puede usar los grupos de S3 Lente de almacenamiento para obtener una visibilidad pormenorizada de grandes buckets compartidos, como los lagos de datos, con el fin de tomar decisiones empresariales mejor fundamentadas. Por ejemplo, puede agilizar la asignación del almacenamiento y optimizar los informes de costes dividiendo el uso del almacenamiento en grupos de objetos específicos para proyectos individuales y centros de costes dentro de un mismo bucket o en varios buckets.

Para usar grupos de S3 Lente de almacenamiento, debe actualizar su panel para usar métricas y recomendaciones avanzadas. Para obtener más información acerca de los grupos de S3 Storage Lens, consulte the section called “Trabajo con grupos de S3 Storage Lens”.

Actualización de un panel de Amazon S3 Storage Lens

Siga los pasos que figuran a continuación para actualizar un panel de Amazon S3 Storage Lens en la consola de Amazon S3.

Paso 1: actualizar el alcance del panel

1. Inicie sesión AWS Management Console Management Console y abra la consola de Amazon S3 en https://console.aws.amazon.com/s3/.
2. En el panel de navegación izquierdo, elija Storage Lens, Dashboards (Storage Lens, paneles).
3. Elija el panel que desea editar y, a continuación, elija Edit (Editar).

   Se abre la página Edit dashboard (Editar panel).

   > **Note**
   >
   > No puede cambiar lo siguiente:
   >
   > - El nombre del panel
   > - La región de origen
   > - El alcance del panel del panel predeterminado, que tiene como alcance el almacenamiento completo de la cuenta

4. (Opcional) En la página de configuración del panel, en la sección General, actualice y agregue etiquetas al panel.

   Puede usar etiquetas a fin de administrar los permisos de su panel y realizar un seguimiento de los costos de S3 Storage Lens. Para obtener más información, consulte Control del acceso mediante etiquetas de recursos en la Guía del usuario de IAM y Etiquetas de asignación de costos generadas por AWS en la Guía del usuario de AWS Billing.

   > **Note**
   >
   > Puede agregar hasta 50 etiquetas a la configuración del panel.

5. En la sección Dashboard scope(Alcance del panel), haga lo siguiente:

   a. Actualice las regiones y los buckets que desea que S3 Storage Lens incluya o excluya en el panel.

> **Note**
>
> - Puede incluir o excluir regiones y buckets. Esta opción está limitada a las regiones solo cuando se crean paneles de nivel de organización en las cuentas de miembros de su organización.
> - Puede elegir hasta 50 buckets para incluir o excluir.

b. Actualice los buckets en las regiones seleccionadas que desea que S3 Storage Lens incluya o excluya. Puede incluir o excluir buckets, pero no ambos. Esta opción no está presente cuando crea paneles de nivel de organización.

Paso 2: actualizar la selección de métricas

1. En la sección Metrics selection (Selección de Métricas), elija el tipo de métricas que desea agregar a este panel.

   - Para incluir métricas gratuitas agregadas en el nivel de bucket y disponibles para consultas durante 14 días, elija Free Metrics (Métricas gratuitas).
   - Para habilitar las métricas avanzadas y otras opciones avanzadas, elija Advanced metrics and recommendations (Métricas y recomendaciones avanzadas). Estas opciones incluyen la agregación de prefijos, la publicación en Amazon CloudWatch y las recomendaciones contextuales avanzadas. Los datos están disponibles para consultas durante 15 meses. Las métricas y recomendaciones avanzadas tienen un costo adicional. Para obtener más información, consulte precios de Amazon S3.

     Para obtener más información sobre las métricas avanzadas y gratuitas, consulte Selección de métricas.

2. En Advanced metrics and recommendations features (Características de métricas y recomendaciones avanzadas), seleccione las opciones que desea habilitar:

   - Advanced metrics (Métricas avanzadas)
   - Publicación de CloudWatch
   - Agregación de prefijos

> **Important**
>
> Si habilita la agregación de prefijos para la configuración de S3 Storage Lens, las métricas de nivel de prefijo no se publicarán en CloudWatch. En CloudWatch, solo se publican métricas de S3 Storage Lens a nivel de bucket, cuenta y organización.

3. Si ha habilitado Advanced metrics (Métricas avanzadas), seleccione las Advanced metrics categories (Categorías de métricas avanzadas) que desea mostrar en el panel de Lente de almacenamiento de S3:

   - Métricas de actividad
   - Detailed status code metrics (Métricas de código de estado detalladas)
   - Advanced cost optimization metrics (Métricas de optimización de costos avanzadas)
   - Advanced data protection metrics (Métricas de protección de datos avanzadas)

   Para obtener más información sobre categorías de métricas, consulte Categorías de métricas. Para obtener una lista completa de métricas, consulte Glosario de métricas de Amazon S3 Storage Lens.

4. Si eligió habilitar la agregación de prefijos, configure lo siguiente:

   a. Elija el tamaño mínimo de umbral de prefijo para este panel.

      Por ejemplo, un umbral de prefijo del 5 por ciento indica que se agregarán los prefijos que componen el 5 por ciento o más del tamaño del almacenamiento total del bucket.

   b. Elija la profundidad del prefijo.

      Esta configuración indica el número máximo de niveles hasta los que se evalúan los prefijos. La profundidad del prefijo debe ser inferior a 10.

   c. Escriba un carácter delimitador de prefijo.

      Este es el valor utilizado para identificar cada nivel de prefijo. El valor predeterminado de Amazon S3 es el carácter `/`, pero su estructura de almacenamiento podría utilizar otros caracteres delimitadores.

(Opcional) Paso 3: exportar métricas para el panel

1. En la sección Metrics Export (Exportación de métricas), para crear una exportación de métricas que se colocará diariamente en un bucket de destino de su elección, elija Enable (Habilitar). Para desactivar la exportación de métricas, elija Disable (Desactivar).

   La exportación de métricas está en formato CSV o Apache Parquet. Representa el mismo alcance de datos que los datos del panel de S3 Storage Lens sin las recomendaciones.

2. Si está habilitado, elija el formato de salida de la exportación diaria de métricas: CSV o Apache Parquet.

   Parquet es un formato de archivo de código abierto para Hadoop que almacena los datos anidados en un formato de columna plano.

3. Elija el bucket de S3 de destino para la exportación de métricas.

   Puede elegir un bucket en la cuenta actual del panel S3 Storage Lens. También puede elegir otra Cuenta de AWS si tiene los permisos del bucket de destino y el ID de la cuenta del propietario del bucket de destino.

4. Elija el bucket de S3 de destino (formato: `s3://`*`bucket-name`*`/`*`prefix`*).

   El bucket debe estar en la región de inicio del panel de Lente de almacenamiento de S3. La consola de S3 le muestra el Destination bucket permission (Permiso del bucket de destino) que agregará Amazon S3 a la política de bucket de destino. Amazon S3 actualiza la política de bucket en el bucket de destino para permitir que S3 coloque datos en ese bucket.

5. (Opcional) Para habilitar el cifrado del lado del servidor para la exportación de métricas, elija Specify an encryption key (Especificar una clave de cifrado). A continuación, elija el tipo de cifrado: Claves administradas de Amazon S3 (SSE-S3) o Clave de AWS Key Management Service (SSE-KMS).

   Puede elegir entre una clave administrada de Amazon S3 (SSE-S3) y una clave de AWS Key Management Service (AWS KMS) (SSE-KMS).

6. (Opcional) Para especificar una clave de AWS KMS, debe elegir una clave KMS o ingresar un nombre de recurso de Amazon (ARN) de clave. En Clave AWS KMS, especifique su clave de KMS de una de las siguientes maneras:

   - Para seleccionar de una lista de claves de KMS disponibles, marque Elija entre sus claves de AWS KMS keys y seleccione su clave de KMS de la lista de claves disponibles.

En esta lista aparecen tanto la Clave administrada de AWS (`aws/s3`) como las claves administradas por el cliente. Para obtener más información acerca de las claves administradas por el cliente, consulte Claves de cliente y claves de AWS en la Guía para desarrolladores de AWS Key Management Service.

> **Note**
>
> La Clave administrada de AWS (`aws/S3`) no es compatible para el cifrado SSE-KMS con Lente de almacenamiento de S3.

- Para introducir el ARN de la clave de KMS, elija Introducir el ARN de la AWS KMS key e introduzca el ARN de la clave de KMS en el campo que aparece.
- Para crear una clave en la consola de AWS KMS, elija Crear una clave de KMS.

  Si elige una clave administrada por el cliente, debe conceder permiso a Lente de almacenamiento de S3 para cifrar la política de claves de AWS KMS. Para obtener más información, consulte Uso de una AWS KMS key para cifrar las exportaciones de métricas.

  Para obtener más información acerca de cómo crear una AWS KMS key, consulte Creación de claves en la AWS Key Management Service Guía para desarrolladores.

7. Elija Guardar cambios.

Para lograr una mayor visibilidad de su almacenamiento, puede crear uno o más grupos de S3 Lente de almacenamiento y asociarlos a su panel. Un grupo de S3 Lente de almacenamiento es un filtro definido y personalizado para objetos basado en prefijos, sufijos, etiquetas de objetos, tamaño de objetos, antigüedad de objetos o una combinación de estos filtros.

Puede usar los grupos de S3 Lente de almacenamiento para obtener una visibilidad pormenorizada de grandes buckets compartidos, como los lagos de datos, con el fin de tomar decisiones empresariales mejor fundamentadas. Por ejemplo, puede agilizar la asignación del almacenamiento y optimizar los informes de costes dividiendo el uso del almacenamiento en grupos de objetos específicos para proyectos individuales y centros de costes dentro de un mismo bucket o en varios buckets.

Para usar grupos de S3 Lente de almacenamiento, debe actualizar su panel para usar métricas y recomendaciones avanzadas. Para obtener más información acerca de los grupos de S3 Storage Lens, consulte the section called “Trabajo con grupos de S3 Storage Lens”.

Deshabilitación o eliminación de paneles de Amazon S3 Storage Lens

Lente de almacenamiento de S3 agrega las métricas y muestra la información en la sección Account snapshot (Instantánea de la cuenta) en la página Buckets de la consola de Amazon S3. S3 Storage Lens también proporciona un panel interactivo que puede utilizar para visualizar información y tendencias, marcar valores atípicos y recibir recomendaciones para optimizar los costes de almacenamiento y aplicar las prácticas recomendadas de protección de datos. El panel tiene opciones de desglose para generar y visualizar información en el nivel de la organización, la cuenta, la Región de AWS, la clase de almacenamiento, el bucket, el prefijo o el grupo de Lente de almacenamiento. También puede enviar una exportación de métricas diaria en formato CSV o Parquet a un bucket de S3.

El panel predeterminado de Amazon S3 Storage Lens es default-account-dashboard. Amazon S3 preconfigura este panel para ayudarle a visualizar información resumida y tendencias para las métricas avanzadas y gratuitas agregadas de toda la cuenta en la consola. No puede modificar el alcance de configuración del panel, pero puede actualizar la selección de métricas de métricas gratuitas a métricas y recomendaciones avanzadas pagadas, configurar la exportación de métricas opcionales o incluso desactivar el panel predeterminado. El panel predeterminado no se puede eliminar.

Puede eliminar o deshabilitar un panel de Amazon S3 Storage Lens desde la consola de Amazon S3. Deshabilitar o eliminar un panel impide que genere métricas en el futuro. Aunque el panel esté deshabilitado, retiene su información de configuración, por lo que se puede reanudar fácilmente cuando se vuelve a habilitar. Un panel desactivado retiene sus datos históricos hasta que ya no esté disponible para consultas.

Los datos para las selecciones de métricas gratuitas se habilitan para consultas durante 14 días y los datos de las selecciones de métricas y recomendaciones avanzadas durante 15 meses.

Deshabilitación de un panel de Amazon S3 Storage Lens dashboard

Para deshabilitar un panel de S3 Storage Lens

1. Inicie sesión AWS Management Console Management Console y abra la consola de Amazon S3 en https://console.aws.amazon.com/s3/.
2. En el panel de navegación izquierdo, elija Storage Lens, Dashboards (Paneles).
3. En la lista Dashboards (Paneles), elija el panel que desea deshabilitar y, a continuación, elija Disable (Deshabilitar) en la parte superior de la lista.

4. En la página de confirmación, confirme que desea deshabilitar el panel escribiendo el nombre del panel en el campo de texto y, a continuación, elija Confirmar.

Eliminación de un panel de Amazon S3 Storage Lens

> **Note**
>
> No puede eliminar el panel predeterminado. No obstante, sí puede deshabilitarlo. Antes de eliminar un panel que ha creado, tenga en cuenta lo siguiente:
>
> - Como alternativa a eliminar un panel, puede deshabilitar el panel para que esté disponible y pueda volver a habilitarse en el futuro. Para obtener más información, consulte Deshabilitación de un panel de Amazon S3 Storage Lens dashboard.
> - Al eliminar el panel, se eliminan todas las opciones de configuración asociadas a él.
> - Cuando elimine un panel, todos los datos de métricas históricas dejarán de estar disponibles. Estos datos históricos se retienen durante 15 meses. Si desea acceder de nuevo a estos datos, cree un panel con el mismo nombre en la misma región de origen que el que se eliminó.

Para eliminar un panel de S3 Storage Lens

1. Inicie sesión en la AWS Management Console y abra la consola de Amazon S3 en https://console.aws.amazon.com/s3/.
2. En el panel de navegación izquierdo, elija Storage Lens, Dashboards (Paneles).
3. En la lista Dashboards (Paneles), elija el panel que desea eliminar y, a continuación, elija Delete (Eliminar) en la parte superior de la lista.
4. En la página Eliminar paneles, confirme que desea eliminar el panel escribiendo el nombre del panel en el campo de texto. A continuación, seleccione Confirm (Confirmar).

Trabajo con AWS Organizations para crear paneles de nivel de organización

Lente de almacenamiento de S3 agrega las métricas y muestra la información en la sección Account snapshot (Instantánea de la cuenta) en la página Buckets de la consola de Amazon S3. S3 Storage Lens también proporciona un panel interactivo que puede utilizar para visualizar información y tendencias, marcar valores atípicos y recibir recomendaciones para optimizar los costes de almacenamiento y aplicar las prácticas recomendadas de protección de datos. El panel tiene

opciones de desglose para generar y visualizar información en el nivel de la organización, la cuenta, la Región de AWS, la clase de almacenamiento, el bucket, el prefijo o el grupo de Lente de almacenamiento. También puede enviar una exportación de métricas diaria en formato CSV o Parquet a un bucket de S3.

El panel predeterminado de Amazon S3 Storage Lens es default-account-dashboard. Amazon S3 preconfigura este panel para ayudarle a visualizar información resumida y tendencias para las métricas avanzadas y gratuitas agregadas de toda la cuenta en la consola. No puede modificar el alcance de configuración del panel, pero puede actualizar la selección de métricas de métricas gratuitas a métricas y recomendaciones avanzadas pagadas, configurar la exportación de métricas opcionales o incluso desactivar el panel predeterminado. El panel predeterminado no se puede eliminar.

También puede crear paneles adicionales de S3 Storage Lens que estén centrados en Regiones de AWS específicas, buckets de S3 u otras Cuentas de AWS de su organización.

Un panel de S3 Lente de almacenamiento proporciona un gran recurso de información sobre su nivel de almacenamiento. Un panel visualiza más de 30 métricas que representan tendencias e información, incluido el resumen del almacenamiento, la rentabilidad, la protección de datos y la actividad.

Amazon S3 Storage Lens se puede utilizar para recopilar métricas de almacenamiento y datos de uso de todas las cuentas que forman parte de la jerarquía de AWS Organizations. Para ello, debe utilizar AWS Organizations y habilitar el acceso de confianza de Lente de almacenamiento de S3 con la cuenta de administración de AWS Organizations.

Cuando el acceso de confianza está habilitado, puede agregar acceso de administrador delegado a las cuentas de la organización. Estas cuentas pueden crear paneles y configuraciones de toda la organización para S3 Storage Lens. Para obtener más información acerca de cómo habilitar el acceso de confianza, consulte Amazon S3 Lens y AWS Organizations en la Guía del usuario de AWS Organizations.

Los siguientes controles de consola solo están disponibles para las cuentas de administración de AWS Organizations.

Habilitación del acceso de confianza para S3 Storage Lens en su organización

La habilitación del acceso de confianza permite a Lente de almacenamiento de Amazon S3 acceder a la jerarquía, la membresía y la estructura de AWS Organizations a través de las operaciones de la API de AWS Organizations. Lente de almacenamiento de S3 se convierte en un servicio de confianza

para toda la estructura de su organización. Puede crear roles vinculados a servicios en cuentas de administración o administrador delegado de su organización cada vez que se cree una configuración de panel.

El rol vinculado a servicios otorga permisos de S3 Storage Lens para describir organizaciones, enumerar cuentas, verificar una lista de acceso al servicio para las organizaciones y obtener administradores delegados para la organización. Esto permite a Lente de almacenamiento de S3 recopilar métricas de actividad y uso de almacenamiento entre cuentas para paneles dentro de las cuentas de su organización.

Para obtener más información, consulte Uso de roles vinculados a servicios para Amazon S3 Storage Lens.

> **Note**
>
> - El acceso de confianza solo se puede habilitar mediante la cuenta de administración.
> - Solo la cuenta de administración y los administradores delegados pueden crear paneles o configuraciones de S3 Storage Lens para su organización.

Para habilitar S3 Storage Lens a fin de que tenga acceso de confianza

1. Inicie sesión AWS Management Console Management Console y abra la consola de Amazon S3 en https://console.aws.amazon.com/s3/.
2. En el panel de navegación izquierdo, elija Storage Lens, Organization settings (Configuración de la organización).
3. En Organizations access (Acceso de organizaciones), elija Edit (Editar).

   Se abrirá la página de Organization access (Acceso a la organización). Aquí puede habilitar el acceso de confianza para S3 Storage Lens. Esto permite a usted y a cualquier otro titular de cuentas, que agregue como administradores delegados, crear paneles para todas las cuentas y el almacenamiento de su organización.

Deshabilitación del acceso de confianza de S3 Storage Lens en su organización

Deshabilitar el acceso de confianza limitará el funcionamiento de Lente de almacenamiento de S3 y solo funcionará en el nivel de cuenta. Cada titular de la cuenta solo podrá ver los beneficios de S3 Storage Lens limitadas al alcance de su cuenta y no a su organización. Los paneles que requieran

acceso de confianza ya no se actualizarán, pero se podrán consultar sus datos históricos según el periodo respectivo en el que los datos están disponibles para consultas.

La eliminación de una cuenta como administrador delegado limita el acceso a las métricas de Lente de almacenamiento de S3 del propietario de la cuenta para que solo funcionen en el nivel de cuenta. Los paneles de la organización que se crearon ya no se actualizarán, pero podrán consultar sus datos históricos según el periodo en el están disponibles para consultas.

> **Note**
>
> - Deshabilitar el acceso de confianza también deshabilita automáticamente todos los paneles de nivel de organización, ya que S3 Storage Lens ya no tendrá acceso de confianza a las cuentas de la organización para recopilar y agregar métricas de almacenamiento.
> - Las cuentas de administración y administrador delegado todavía pueden ver los datos históricos de estos paneles desactivados y pueden consultar estos datos mientras estén disponibles.

Para deshabilitar el acceso de confianza para S3 Storage Lens

1. Inicie sesión AWS Management Console Management Console y abra la consola de Amazon S3 en https://console.aws.amazon.com/s3/.
2. En el panel de navegación izquierdo, elija Storage Lens, Organization settings (Configuración de la organización).
3. En Organizations access (Acceso de organizaciones), elija Edit (Editar).

   Se abrirá la página de Organization access (Acceso a la organización). Aquí puede deshabilitar el acceso de confianza para S3 Storage Lens.

Registro de administradores delegados para S3 Storage Lens

Después de habilitar el acceso de confianza, puede registrar el acceso del administrador delegado a cuentas de su organización. Cuando se registra una cuenta como administrador delegado, la cuenta recibe autorización para acceder a todas las operaciones de la API de solo lectura de AWS Organizations. Esto proporciona visibilidad a los miembros y estructuras de su organización para que puedan crear paneles de S3 Storage Lens en su nombre.

A fin de registrar administradores delegados para S3 Storage Lens

1. Inicie sesión AWS Management Console Management Console y abra la consola de Amazon S3 en https://console.aws.amazon.com/s3/.
2. En el panel de navegación izquierdo, elija Storage Lens, Organization settings (Configuración de la organización).
3. En la sección delegated access (acceso delegado), en Accounts (Cuentas), elija Add account (Agregar cuenta).

   Se abrirá la página Delegated admin access (Acceso de administrador delegado). Aquí puede agregar un ID de Cuenta de AWS como administrador delegado con el fin de crear paneles de nivel de organización para todas las cuentas y el almacenamiento de su organización.

Anular el registro de administradores delegados para S3 Storage Lens

Puede anular el registro del acceso de administrador delegado a las cuentas de su organización. Cuando se anula el registro de una cuenta como administrador delegado, la cuenta pierde la autorización para acceder a todas las operaciones de la API de solo lectura de AWS Organizations que proporcionan visibilidad a los miembros y las estructuras de su organización.

> **Note**
>
> - Anular el registro de un administrador delegado también deshabilita automáticamente todos los paneles de nivel de organización creados por el administrador delegado.
> - Las cuentas de administrador delegado todavía pueden ver los datos históricos de estos paneles desactivados de acuerdo el respectivo periodo en el que los datos están disponibles para consultas.

Para anular el registro de cuentas para el acceso de administrador delegado

1. Inicie sesión AWS Management Console Management Console y abra la consola de Amazon S3 en https://console.aws.amazon.com/s3/.
2. En el panel de navegación izquierdo, elija Storage Lens, Organization settings (Configuración de la organización).
3. En la sección Accounts with delegated access (Cuentas con acceso delegado), elija el ID de cuenta para el que desea anular el registro y, a continuación, elija Remove (Eliminar).

Ejemplos de Lente de almacenamiento de Amazon S3 con la AWS CLI

Lente de almacenamiento de S3 agrega las métricas y muestra la información en la sección Account snapshot (Instantánea de la cuenta) en la página Buckets de la consola de Amazon S3. Lente de almacenamiento de S3 también proporciona un panel interactivo que puede utilizar para visualizar información y tendencias, marcar valores atípicos y recibir recomendaciones para optimizar los costos de almacenamiento y aplicar las prácticas recomendadas de protección de datos. El panel tiene opciones de desglose para generar y visualizar información en el nivel de la organización, la cuenta, la Región de AWS, la clase de almacenamiento, el bucket, el prefijo o el grupo de Storage Lens. También puede enviar una exportación de métricas diaria en formato CSV o Parquet a un bucket de S3. Para obtener más información, consulte Evaluación de la actividad y el uso de almacenamiento con Amazon S3 Storage Lens.

En los siguientes ejemplos, se muestra cómo puede utilizar Lente de almacenamiento de S3 con AWS Command Line Interface.

Temas

- Archivos auxiliares para utilizar Amazon S3 Storage Lens
- Uso de las configuraciones de la Lente de almacenamiento de Amazon S3 con la AWS CLI
- Uso de la Lente de almacenamiento de Amazon S3 con AWS Organizations mediante la AWS CLI

Archivos auxiliares para utilizar Amazon S3 Storage Lens

Utilice los siguientes archivos JSON y las entradas de claves de sus ejemplos.

Ejemplo de configuración de S3 Storage Lens en JSON

Example **`config.json`**

El archivo `config.json` contiene detalles de una configuración de métricas y recomendaciones avanzadas de organización de Lente de almacenamiento de S3. Para utilizar el ejemplo siguiente, sustituya *`user input placeholders`* con su propia información.

> **Note**
>
> Se aplican cargos adicionales a las métricas y recomendaciones avanzadas. Para obtener más información, consulte Métricas y recomendaciones avanzadas.

```
{
  "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
    }
  }
}
```

Ejemplo de configuración de S3 Storage Lens con grupos de Storage Lens en JSON

Example **`config.json`**

El archivo `config.json` contiene los detalles que desea aplicar a la configuración de Storage Lens al usar grupos de Storage Lens. Para utilizar el ejemplo, sustituya *`user input placeholders`* por su propia información.

Para asociar todos los grupos de Storage Lens a su panel de control, actualice la configuración de Storage Lens con la siguiente sintaxis:

```
{
  "Id": "ExampleS3StorageLensConfiguration",
  "AccountLevel": {
```

```
    "ActivityMetrics": {
      "IsEnabled":true
    },
    "AdvancedCostOptimizationMetrics": {
      "IsEnabled":true
    },
    "AdvancedDataProtectionMetrics": {
      "IsEnabled":true
    },
    "BucketLevel": {
      "ActivityMetrics": {
      "IsEnabled":true
      },
    "StorageLensGroupLevel": {},
  "IsEnabled": true
}
```

Para incluir solo dos grupos de Storage Lens en la configuración del panel de Storage Lens (*slg-1* y *slg-2*), utilice la siguiente sintaxis:

```
{
  "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
}
```

Para impedir que solo determinados grupos de Storage Lens se adjunten a la configuración de su panel, utilice la siguiente sintaxis:

```
{
  "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
}
```

Etiquetas de configuración de ejemplo de S3 Storage Lens

Example **`tags.json`**

El archivo `tags.json` contiene las etiquetas que desea aplicar a la configuración de Lente de almacenamiento de S3. Para utilizar este ejemplo, reemplace los *`user input placeholders`* con su propia información.

```
[
    {
        "Key": "key1",
```

```
        "Value": "value1"
    },
    {
        "Key": "key2",
        "Value": "value2"
    }
]
```

Permisos de IAM de configuración de ejemplo de S3 Storage Lens

Example **`permissions.json`**: nombre de panel específico

Esta política de ejemplo muestra un archivo de `permissions.json` de IAM de Lente de almacenamiento de S3 con un nombre de panel específico designado. Sustituya *`value1`*, *`us-east-1`*, *`your-dashboard-name`* y *`example-account-id`* por sus propios valores.

```
{
    "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`**: nombre de panel no específico

Esta política de ejemplo muestra un archivo de `permissions.json` de IAM de Lente de almacenamiento de S3 sin un nombre de panel específico designado. Reemplace *`value1`*, *`us-east-1`* y *`example-account-id`* por sus propios valores.

```
{
    "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/*"
        }
    ]
}
```

Uso de las configuraciones de la Lente de almacenamiento de Amazon S3 con la AWS CLI

Puede utilizar la AWS CLI para mostrar, crear, eliminar obtener, etiquetar y actualizar las configuraciones de Lente de almacenamiento de S3. En los siguientes ejemplos se utilizan los archivos JSON auxiliares para las entradas de claves. Para utilizar estos ejemplos, sustituya *`user input placeholders`* por su propia información.

Crear una configuración de Lente de almacenamiento de S3

Example Crear una configuración de Lente de almacenamiento de S3

```
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
```

Crear una configuración de Lente de almacenamiento de S3 sin etiquetas

Example Crear una configuración de Lente de almacenamiento de S3 sin etiquetas

```
aws s3control put-storage-lens-configuration --account-id=222222222222 --config-
id=your-configuration-id --region=us-east-1 --storage-lens-configuration=file://./
config.json
```

Obtener una configuración de S3 Storage Lens

Example Obtener una configuración de S3 Storage Lens

```
aws s3control get-storage-lens-configuration --account-id=222222222222 --config-
id=your-configuration-id --region=us-east-1
```

Enumeración de las configuraciones de Lente de almacenamiento de S3 sin un siguiente token

Example Enumeración de las configuraciones de Lente de almacenamiento de S3 sin un siguiente token

```
aws s3control list-storage-lens-configurations --account-id=222222222222 --region=us-
east-1
```

Enumeración de las configuraciones de S3 Storage Lens

Example Enumeración de las configuraciones de S3 Storage Lens

```
aws s3control list-storage-lens-configurations --account-id=222222222222 --region=us-
east-1 --next-token=abcdefghij1234
```

Eliminar una configuración de S3 Storage Lens

Example Eliminar una configuración de S3 Storage Lens

```
aws s3control delete-storage-lens-configuration --account-id=222222222222 --region=us-
east-1 --config-id=your-configuration-id
```

Agregar etiquetas a una configuración de Lente de almacenamiento de S3

Example Agregar etiquetas a una configuración de Lente de almacenamiento de S3

```
aws s3control put-storage-lens-configuration-tagging --account-id=222222222222 --
region=us-east-1 --config-id=your-configuration-id --tags=file://./tags.json
```

Obtener etiquetas para una configuración de S3 Storage Lens

Example Obtener etiquetas para una configuración de S3 Storage Lens

```
aws s3control get-storage-lens-configuration-tagging --account-id=222222222222 --
region=us-east-1 --config-id=your-configuration-id
```

Eliminar etiquetas para una configuración de S3 Storage Lens

Example Eliminar etiquetas para una configuración de S3 Storage Lens

```
aws s3control delete-storage-lens-configuration-tagging --account-id=222222222222 --
region=us-east-1 --config-id=your-configuration-id
```

Uso de la Lente de almacenamiento de Amazon S3 con AWS Organizations mediante la AWS CLI

Utilice Amazon S3 Storage Lens para recopilar métricas de almacenamiento y datos de uso de todas las cuentas que forman parte de la jerarquía de AWS Organizations. Para obtener más información, consulte Uso de Amazon S3 Storage Lens con AWS Organizations.

Habilitar el acceso de confianza de Organizations para S3 Storage Lens

Example Habilitar el acceso de confianza de Organizations para S3 Storage Lens

```
aws organizations enable-aws-service-access --service-principal storage-
lens.s3.amazonaws.com
```

Deshabilitar el acceso de confianza de Organizations para S3 Storage Lens

Example Deshabilitar el acceso de confianza de Organizations para S3 Storage Lens

```
aws organizations disable-aws-service-access --service-principal storage-
lens.s3.amazonaws.com
```

Registrar administradores delegados de Organizations para S3 Storage Lens

Example Registrar administradores delegados de Organizations para S3 Storage Lens

Para usar este ejemplo, sustituya *111122223333* por el ID de Cuenta de AWS correspondiente.

```
aws organizations register-delegated-administrator --service-principal storage-
lens.s3.amazonaws.com --account-id 111122223333
```

Anulación del registro de administradores delegados de Organizations para S3 Storage Lens

Example Anulación del registro de administradores delegados de Organizations para S3 Storage Lens

Para usar este ejemplo, sustituya *111122223333* por el ID de Cuenta de AWS correspondiente.

```
aws organizations deregister-delegated-administrator --service-principal storage-lens.s3.amazonaws.com --account-id 111122223333
```

Ejemplos de Amazon S3 Storage Lens en los que se utiliza SDK para Java

S3 Storage Lens agrega las métricas y muestra la información en la sección Instantánea de la cuenta en la página Buckets de la consola de Amazon S3. S3 Storage Lens también proporciona un panel interactivo que puede utilizar para visualizar información y tendencias, marcar valores atípicos y recibir recomendaciones para optimizar los costes de almacenamiento y aplicar las prácticas recomendadas de protección de datos. El panel tiene opciones de desglose para generar y visualizar información en el nivel de la organización, la cuenta, la Región de AWS, la clase de almacenamiento, el bucket, el prefijo o el grupo de Storage Lens. También puede enviar una exportación de métricas diaria en formato CSV o Parquet a un bucket de S3. Para obtener más información, consulte Evaluación de la actividad y el uso de almacenamiento con Amazon S3 Storage Lens.

En los siguientes ejemplos, se muestra cómo puede utilizar Lente de almacenamiento de S3 con AWS SDK for Java.

Temas

- Uso de configuraciones de Amazon S3 Storage Lens mediante SDK para Java

Uso de configuraciones de Amazon S3 Storage Lens mediante SDK para Java

Puede utilizar SDK para Java a fin de enumerar, crear, obtener y actualizar las configuraciones de S3 Storage Lens. En los siguientes ejemplos se utilizan los archivos JSON auxiliares para las entradas de claves.

Temas

- Crear y actualizar una configuración de S3 Storage Lens
- Eliminar una configuración de S3 Storage Lens
- Obtener una configuración de S3 Storage Lens

- Enumeración de las configuraciones de S3 Storage Lens
- Agregar etiquetas a una configuración de S3 Storage Lens
- Obtener etiquetas para una configuración de S3 Storage Lens
- Eliminar etiquetas para una configuración de S3 Storage Lens
- Actualizar la configuración predeterminada de S3 Storage Lens con métricas y recomendaciones avanzadas
- Asociar un grupo de Storage Lens a un panel de S3 Storage Lens
- Uso de la Amazon S3 Storage Lens con ejemplos de AWS Organizations mediante el SDK para Java

Crear y actualizar una configuración de S3 Storage Lens

Example Crear y actualizar una configuración de 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();
        }
    }
}
```

Eliminar una configuración de S3 Storage Lens

Example Eliminar una configuración de 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();
        }
    }
}
```

Obtener una configuración de S3 Storage Lens

Example Obtener una configuración de 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();
        }
    }
}
```

Enumeración de las configuraciones de S3 Storage Lens

Example Enumeración de las configuraciones de 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();
        }
    }
}
```

Agregar etiquetas a una configuración de S3 Storage Lens

Example Agregar etiquetas a una configuración de 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();
        }
    }
}
```

Obtener etiquetas para una configuración de S3 Storage Lens

Example Obtener etiquetas para una configuración de 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();
        }
    }
}
```

Eliminar etiquetas para una configuración de S3 Storage Lens

Example Eliminar etiquetas para una configuración de 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();
        }
    }
}
```

Actualizar la configuración predeterminada de S3 Storage Lens con métricas y recomendaciones avanzadas

Example Actualizar la configuración predeterminada de S3 Storage Lens con métricas y recomendaciones avanzadas

```
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**
>
> Se aplican cargos adicionales a las métricas y recomendaciones avanzadas. Para obtener más información, consulte Métricas y recomendaciones avanzadas.

Asociar un grupo de Storage Lens a un panel de S3 Storage Lens

Example Asociar todos los grupos de Storage Lens a un panel

En el siguiente ejemplo de SDK para Java, se asocian todos los grupos de Storage Lens de la cuenta *111122223333* al panel *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 Asociar dos grupos de Storage Lens a un panel

En el siguiente ejemplo de AWS SDK for Java, se asocian dos grupos de Storage Lens (*StorageLensGroupName1* y *StorageLensGroupName2*) al panel *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 Asociar todos los grupos de Storage Lens a exclusiones

El siguiente ejemplo de SDK para Java asocia todos los grupos de Storage Lens al panel *ExampleDashboardConfigurationID*, excepto los dos especificados (*StorageLensGroupName1* y *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();
        }
    }
}
```

Uso de la Amazon S3 Storage Lens con ejemplos de AWS Organizations mediante el SDK para Java

Utilice Amazon S3 Storage Lens para recopilar métricas de almacenamiento y datos de uso de todas las cuentas que forman parte de la jerarquía de AWS Organizations. Para obtener más información, consulte Uso de Amazon S3 Storage Lens con AWS Organizations.

Temas

- Habilitar el acceso de confianza de Organizations para S3 Storage Lens
- Deshabilitar el acceso de confianza de Organizations para S3 Storage Lens
- Registrar administradores delegados de Organizations para S3 Storage Lens
- Anulación del registro de administradores delegados de Organizations para S3 Storage Lens

Habilitar el acceso de confianza de Organizations para S3 Storage Lens

Example Habilitar el acceso de confianza de Organizations para 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.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();
        }
 }
}
```

Deshabilitar el acceso de confianza de Organizations para S3 Storage Lens

Example Deshabilitar el acceso de confianza de Organizations para 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.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();
        }
 }
}
```

Registrar administradores delegados de Organizations para S3 Storage Lens

Example Registrar administradores delegados de Organizations para 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();
        }
 }
}
```

Anulación del registro de administradores delegados de Organizations para S3 Storage Lens

Example Anulación del registro de administradores delegados de Organizations para 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();
        }
 }
}
```

Trabajo con grupos de S3 Storage Lens

Un grupo de Amazon S3 Storage Lens agrega métricas mediante filtros personalizados en función de metadatos de objeto. Los grupos de Storage Lens le ayudan a desglosar las características de sus datos, como la distribución de los objetos por edad, sus tipos de archivos más comunes, etc. Por ejemplo, puede filtrar las métricas por etiqueta de objeto para identificar sus conjuntos de datos de crecimiento más rápido o visualizar su almacenamiento en función del tamaño y la edad del objeto para fundamentar su estrategia de archivado de almacenamiento. Como resultado, los grupos de Amazon S3 Storage Lens le ayudan a comprender y optimizar mejor su almacenamiento de S3.

Cuando utiliza grupos de Storage Lens, puede analizar y filtrar las métricas de S3 Storage Lens utilizando metadatos de objetos, como prefijos, sufijos, etiquetas de objetos, tamaño de objetos o edad de objetos. También puede aplicar una combinación de estos filtros. Después de asociar su grupo de Storage Lens al panel de S3 Storage Lens, podrá ver las métricas de S3 Storage Lens agregadas por los grupos de Amazon S3 Storage Lens directamente en su panel.

Por ejemplo, también puede filtrar sus métricas por tamaño de objeto o por franjas de edad para determinar qué parte de su almacenamiento está formada por objetos pequeños. A continuación, puede utilizar esta información con S3 Intelligent-Tiering o S3 Lifecycle para realizar la transición de los objetos pequeños a diferentes clases de almacenamiento con el fin de optimizar los costes y el almacenamiento.

Temas

Cómo funcionan los grupos de S3 Storage Lens

Puede usar los grupos de Lente de almacenamiento para agregar métricas mediante filtros personalizados en función de los metadatos de objetos. Al definir un filtro personalizado, puede usar prefijos, sufijos, etiquetas de objetos, tamaños de objetos, antigüedad de objetos o una combinación de estos filtros personalizados. Durante la creación del grupo de Lente de almacenamiento, también puede incluir un filtro único o varias condiciones de filtro. Para especificar varias condiciones de filtro, utilice los operadores lógicos `And` o `Or`.

Al crear y configurar un grupo de Lente de almacenamiento, el propio grupo de Lente de almacenamiento actúa como un filtro personalizado en el panel al que se asocia el grupo. A continuación, en su panel, puede utilizar el filtro de grupo de Lente de almacenamiento para obtener métricas de almacenamiento en función del filtro personalizado que haya definido en el grupo.

Para ver los datos de su grupo de Lente de almacenamiento en el panel Lente de almacenamiento de S3, debe asociar el grupo al panel una vez creado el grupo. Cuando el grupo de Lente de almacenamiento esté asociado al panel de Lente de almacenamiento, este recopilará las métricas de uso del almacenamiento en un plazo de 48 horas. A continuación, puede visualizar estos datos en el panel de Lente de almacenamiento o exportarlos mediante una exportación de métricas. Si olvida asociar un grupo de Lente de almacenamiento a un panel, los datos del grupo de Lente de almacenamiento no se capturarán ni se mostrarán en ningún lugar.

> **Note**
>
> - Cuando crea un grupo de S3 Lente de almacenamiento, está creando un recurso de AWS. Por lo tanto, cada grupo de Lente de almacenamiento tiene su propio nombre de recurso de Amazon (ARN), que puede especificar al asociarlo o excluirlo de un panel de S3 Lente de almacenamiento.
> - Si su grupo de Lente de almacenamiento no está asociado a un panel, no incurrirá en ningún cargo adicional por crear un grupo de Lente de almacenamiento.
> - S3 Lente de almacenamiento agrega las métricas de uso para un objeto en todos los grupos de Lente de almacenamiento coincidentes. Por lo tanto, si un objeto cumple las

condiciones de filtrado de dos o más grupos de Lente de almacenamiento, verá recuentos repetidos para el mismo objeto a lo largo de su uso de almacenamiento.

Puede crear un grupo de Lente de almacenamiento a nivel de cuenta de una región de origen específica (de la lista de Regiones de AWS admitidas). A continuación, puede asociar su grupo de Lente de almacenamiento a varios paneles de Lente de almacenamiento, siempre que los paneles se encuentren en la misma Cuenta de AWS y región de origen. Puede crear hasta 50 grupos de Lente de almacenamiento por región de origen en cada Cuenta de AWS.

Puede crear y administrar los grupos de S3 Storage Lens mediante la consola de Amazon S3, AWS Command Line Interface, (AWS CLI), los SDK de AWS o la API de REST de Amazon S3.

Temas

Visualización de las métricas agregadas del grupo de Lente de almacenamiento

Puede ver las métricas agregadas de sus grupos de Lente de almacenamiento asociando los grupos a un panel. Los grupos de Lente de almacenamiento que desee asociar deben encontrarse en la región de origen designada en la cuenta del panel.

Para asociar un grupo de Lente de almacenamiento a un panel, debe especificar el grupo en la sección Agregación de grupos de Lente de almacenamiento de la configuración del panel. Si tiene varios grupos de Lente de almacenamiento, puede filtrar los resultados de la Agregación de grupos de Lente de almacenamiento para incluir o excluir solo los grupos que desee. Para obtener más información acerca de cómo asociar grupos a sus paneles, consulte the section called “Asociar o eliminar un grupo de Storage Lens”.

Una vez que haya asociado los grupos, verá los datos de agregación de grupos de Lente de almacenamiento adicionales en el panel en un plazo de 48 horas.

> **Note**
>
> Para ver las métricas agregadas para su grupo de Lente de almacenamiento, debe asociar el grupo a un panel de S3 Lente de almacenamiento.

Permisos de grupos de Storage Lens

Los grupos de Storage Lens requieren determinados permisos en AWS Identity and Access Management (IAM) para autorizar el acceso a las acciones del grupo de S3 Storage Lens. Para conceder estos permisos, puede utilizar una política de IAM basada en identidades. Para asociar esta política a usuarios, grupos o roles de IAM para concederles permisos. Estos permisos pueden incluir la posibilidad de crear o eliminar grupos de Lente de almacenamiento, ver sus configuraciones o administrar sus etiquetas.

El rol o usuario de IAM al que concede permisos debe pertenecer a la cuenta que creó o ser propietaria del grupo de Lente de almacenamiento.

Para usar los grupos de Lente de almacenamiento y ver las métricas de los grupos de Lente de almacenamiento, debe tener primero los permisos adecuados para usar S3 Lente de almacenamiento. Para obtener más información, consulte the section called “Permisos de S3 Storage Lens”.

Para crear y administrar grupos de S3 Lente de almacenamiento, debe tener los siguientes permisos de IAM, en función de las acciones que desee realizar:

Acción	Permisos de IAM
Crear un nuevo grupo de Lente de almacenamiento	`s3:CreateStorageLensGroup`
Crear un nuevo grupo de Lente de almacenamiento con etiquetas	`s3:CreateStorageLensGroup` , `s3:TagResource`
Actualizar un grupo de Lente de almacenamiento existente	`s3:UpdateStorageLensGroup`
Devolver los detalles de una configuración de grupo de Lente de almacenamiento	`s3:GetStorageLensGroup`

Acción	Permisos de IAM
Enumerar todos los grupos de Lente de almacenamiento de su región de origen	`s3:ListStorageLensGroups`
Eliminar un grupo de Lente de almacenamiento	`s3:DeleteStorageLensGroup`
Enumerar las etiquetas que se agregaron a su grupo de Lente de almacenamiento	`s3:ListTagsForResource`
Agregar o actualizar una etiqueta de grupo de Lente de almacenamiento para un grupo de Lente de almacenamiento existente	`s3:TagResource`
Eliminar una etiqueta de un grupo de Lente de almacenamiento	`s3:UntagResource`

Este es un ejemplo de cómo configurar la política de IAM en la cuenta que crea el grupo de Lente de almacenamiento. Para usar esta política, sustituya *`us-east-1`* por la región de origen en la que se encuentra su grupo de Lente de almacenamiento. Sustituya *`111122223333`* por su ID Cuenta de AWS, y *`example-storage-lens-group`* por el nombre de su grupo de Lente de almacenamiento. Para aplicar estos permisos a todos los grupos de Lente de almacenamiento, sustituya *`example-storage-lens-group`* por un *`*`*.

```
{
    "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"
        }
    ]
}
```

Para obtener más información acerca del uso de S3 Storage Lens, consulte Permisos de Amazon S3 Storage Lens. Para obtener más información sobre el lenguaje de la política de IAM, consulte Políticas y permisos en Amazon S3.

Configuración de grupos de Lente de almacenamiento

Nombre de grupo de S3 Lente de almacenamiento

Recomendamos dar a los grupos de Lente de almacenamiento nombres que indiquen su finalidad para que pueda determinar con facilidad qué grupos desea asociar a sus paneles. Para asociar un grupo de Lente de almacenamiento a un panel, debe especificar el grupo en la sección Agregación de grupos de Lente de almacenamiento de la configuración del panel.

Los nombres de grupos de Storage Lens deben ser únicos dentro de la cuenta. No deben contener más de 64 caracteres, y pueden contener únicamente letras (a-z, A-Z), números (0-9), guiones (-) y guiones bajos (_).

Región de origen

La región de origen es la Región de AWS donde se crea y mantiene su grupo de Lente de almacenamiento. Su grupo de Lente de almacenamiento se crea en la misma región de origen que su panel de Lente de almacenamiento de Amazon S3. La configuración y las métricas del grupo de Lente de almacenamiento también se almacenan en esta región. Puede crear hasta 50 grupos de Lente de almacenamiento en una región de origen.

Después de crear su grupo de Lente de almacenamiento, no puede editar la región de origen.

Ámbito

Para incluir objetos en su grupo de Lente de almacenamiento, deben estar dentro del ámbito de su panel de Lente de almacenamiento de Amazon S3. El ámbito de su panel de Lente de almacenamiento viene determinado por los buckets que incluyó en el Ámbito del panel de su configuración del panel de S3 Lente de almacenamiento.

Puede usar diferentes filtros para sus objetos con el fin de definir el ámbito de su grupo de Lente de almacenamiento. Para ver estas métricas de grupo de Lente de almacenamiento en su panel de S3

Lente de almacenamiento, los objetos deben coincidir con los filtros que incluye en sus grupos de Lente de almacenamiento. Por ejemplo, supongamos que su grupo de Lente de almacenamiento incluye objetos con el prefijo `marketing` y el sufijo `.png`, pero ningún objeto cumple esos criterios. En este caso, no se generarán las métricas de este grupo de Lente de almacenamiento en su exportación de métricas diaria y no se mostrará ninguna métrica para este grupo en su panel.

Filtros

Puede usar los siguientes filtros en un grupo de S3 Lente de almacenamiento:

- Prefijos: especifica el prefijo de los objetos incluidos, que es una cadena de caracteres al principio del nombre de la clave de objeto. Por ejemplo, un valor de `images` para el filtro de Prefijos incluye los objetos con cualquiera de los siguientes prefijos: `images/`, `images-marketing` y `images/production`. La longitud máxima de un prefijo es de 1024 bytes.
- Sufijos: especifica el sufijo de los objetos incluidos (por ejemplo, `.png`, `.jpeg` o `.csv`). La longitud máxima de un sufijo es de 1024 bytes.
- Etiquetas de objetos: especifica la lista de etiquetas de objetos por las que desea filtrar. Una clave de etiqueta no puede superar los 128 caracteres Unicode y el valor de una etiqueta no puede superar los 256 caracteres Unicode. Tenga en cuenta que si el campo de valor de la etiqueta del objeto se deja vacío, los grupos de lentes de almacenamiento de S3 solo harán coincidir el objeto con otros objetos que también tengan valores de etiqueta vacíos.
- Antigüedad: especifica el rango de antigüedad de los objetos incluidos en días. Solo se admiten números enteros.
- Tamaño: especifica el rango de tamaños de objeto de los objetos incluidos en bytes. Solo se admiten números enteros. El valor máximo permitido es de 5 TB.

Etiquetas de objetos del grupo de Lente de almacenamiento

Puede crear un grupo de Lente de almacenamiento que incluya hasta 10 filtros de etiquetas de objetos. El siguiente ejemplo incluye dos pares clave-valor de etiquetas de objetos como filtros para un grupo de Lente de almacenamiento denominado *`Marketing-Department`*. Para usar este ejemplo, sustituya *`Marketing-Department`* por el nombre del grupo y reemplace*`object-tag-key-1`*, y así sucesivamente*`object-tag-value-1`*, por los pares clave-valor de etiquetas de objeto por los que desee filtrar.

```
{
    "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"
                }
            ]
    }
}
```

Operadores lógicos (**`And`** o **`Or`**)

Para incluir varias condiciones de filtro en su grupo de Lente de almacenamiento, puede utilizar operadores lógicos (`And` o `Or`). En el siguiente ejemplo, el grupo de Lente de almacenamiento *`Marketing-Department`* tiene un operador `And` que contiene los filtros `Prefix`, `ObjectAge` y `ObjectSize`. Como se utiliza un operador `And`, solo los objetos que coincidan con todas estas condiciones de filtro se incluirán en el ámbito del grupo de Lente de almacenamiento.

Para usar este ejemplo, sustituya los *`user input placeholders`* por los valores por los que desee filtrar.

```
{
    "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 desea incluir objetos que cumplan alguna de las condiciones de los filtros, sustituya el operador lógico `And` por el operador lógico `Or` de este ejemplo.

Etiquetas de recursos de AWS

Cada grupo de S3 Lente de almacenamiento se considera un recurso de AWS con su propio nombre de recurso de Amazon (ARN). Por lo tanto, al configurar su grupo de Lente de almacenamiento, si lo desea, puede añadir etiquetas de recursos de AWS al grupo. Puede añadir un máximo de 50 etiquetas a cada grupo de Lente de almacenamiento. Para crear un grupo de Lente de almacenamiento con etiquetas, debe tener los permisos `s3:CreateStorageLensGroup` y `s3:TagResource`.

Puede usar etiquetas de recursos de AWS para clasificar los recursos por departamento, línea de negocio o proyecto. Esto resulta útil cuando se tienen muchos recursos del mismo tipo. Al aplicar etiquetas, puede identificar rápidamente un grupo de Lente de almacenamiento específico en función de las etiquetas que le haya asignado. También puede utilizar etiquetas para realizar un seguimiento de los costes y asignarlos.

Además, cuando agrega una etiqueta de recurso de AWS a su grupo de Lente de almacenamiento, activa el control de acceso basado en atributos (ABAC). ABAC es una estrategia de autorización que define permisos basados en atributos, en este caso, etiquetas. También puede utilizar las condiciones que especifican etiquetas de recursos en sus políticas de IAM para controlar el acceso a los recursos de AWS.

Puede editar las claves y los valores de las etiquetas y también puede eliminar etiquetas de un recurso en cualquier momento. Además, debe tener en cuenta las siguientes limitaciones:

- Las claves y los valores de las etiquetas distinguen entre mayúsculas y minúsculas.
- Si añade una etiqueta con la misma clave que una etiqueta existente en ese recurso, el nuevo valor sobrescribirá al antiguo.
- Si elimina un recurso, también se eliminará cualquier etiqueta asignada a dicho recurso.

- No incluya datos privados o confidenciales en sus etiquetas de recursos de AWS.
- No se admiten etiquetas del sistema (o las etiquetas con claves de etiquetas que comiencen por `aws:`).
- La longitud de cada clave de etiqueta no puede superar los 128 caracteres. La longitud del valor de cada etiqueta no puede superar los 256 caracteres.

Exportación de métricas de grupos de Lente de almacenamiento

Las métricas del grupo de Lente de almacenamiento de S3 se incluyen en la Exportación de métricas de Lente de almacenamiento de Amazon S3 para el panel al que está asociado el grupo de Lente de almacenamiento. Para obtener información general acerca de la característica de exportación de métricas de Lente de almacenamiento, consulte Visualización de las métricas de Amazon S3 Storage Lens mediante una exportación de datos.

La exportación de métricas para los grupos de Lente de almacenamiento incluye las métricas de S3 Lente de almacenamiento que estén incluidas en el panel al que asoció el grupo de Lente de almacenamiento. La exportación también incluye datos de métricas adicionales para los grupos de Lente de almacenamiento.

Una vez creado su grupo de Lente de almacenamiento, la exportación de métricas se envía a diario al bucket que seleccionó al configurar la exportación de métricas para el panel al que está asociado su grupo. Puede tardar hasta 48 horas en recibir la primera exportación de métricas.

Para generar métricas en la exportación diaria, los objetos deben coincidir con los filtros que incluya en sus grupos de Lente de almacenamiento. Si ningún objeto coincide con los filtros que ha incluido en su grupo de Lente de almacenamiento, no se generará ninguna métrica. Sin embargo, si un objeto coincide con dos o más grupos de Lente de almacenamiento, el objeto aparece por separado para cada grupo cuando se muestra en la exportación de métricas.

Para identificar las métricas de los grupos de Lente de almacenamiento, busque uno de los siguientes valores en la columna `record_type` de la exportación de métricas de su panel:

- `STORAGE_LENS_GROUP_BUCKET`
- `STORAGE_LENS_GROUP_ACCOUNT`

La columna `record_value` muestra el ARN del recurso para el grupo de Lente de almacenamiento (por ejemplo, `arn:aws:s3:`*`us-east-1`*`:`*`111122223333`*`:storage-lens-group/`*`Marketing-Department`*).

Uso de grupos de Storage Lens

Los grupos de Amazon S3 Storage Lens agregan métricas mediante filtros personalizados basados en metadatos de objetos. Puede analizar y filtrar las métricas de S3 Storage Lens mediante prefijos, sufijos, etiquetas de objetos, tamaño de objetos y edad de objetos. Con los grupos de Amazon S3 Storage Lens, también puede clasificar su uso dentro de buckets de Amazon S3 y entre ellos. Como resultado, podrá comprender y optimizar mejor su almacenamiento de S3.

Para empezar a visualizar los datos de un grupo de Storage Lens, primero debe asociar el grupo de Storage Lens a un panel de S3 Storage Lens. Si necesita administrar los grupos de Storage Lens en el panel, puede editar la configuración del panel. Para comprobar qué grupos de Storage Lens están en su cuenta, puede incluirlos en una lista. Para comprobar qué grupos de Storage Lens están asociados a su panel, siempre puede consultar la pestaña Grupos de Storage Lens en el panel. Para revisar o actualizar el ámbito de un grupo de Storage Lens existente, puede ver sus detalles. También puede eliminar permanentemente un grupo de Storage Lens.

Para administrar los permisos, puede crear y agregar etiquetas de recursos de AWS definidas por el usuario a sus grupos de Storage Lens. Puede usar etiquetas de recursos de AWS para clasificar recursos según el departamento, la línea de negocio o el proyecto. Esto resulta útil cuando se tienen muchos recursos del mismo tipo. Al aplicar etiquetas, puede identificar rápidamente un grupo de Storage Lens específico en función de las etiquetas que le haya asignado.

Además, cuando agrega una etiqueta de recurso de AWS a su grupo de Storage Lens, activa el control de acceso basado en atributos (ABAC). ABAC es una estrategia de autorización que define permisos basados en atributos, en este caso, etiquetas. También puede utilizar las condiciones de sus políticas de IAM para controlar el acceso a los recursos de AWS.

Temas

Creación de un grupo de Lente de almacenamiento

En los siguientes ejemplos se muestra cómo crear un grupo de Lente de almacenamiento de Amazon S3 mediante la consola de Amazon S3, AWS Command Line Interface (AWS CLI) yAWS SDK for Java.

Uso de la consola de S3

Para crear un grupo de Lente de almacenamiento

1. Inicie sesión AWS Management Console Management Console y abra la consola de Amazon S3 en https://console.aws.amazon.com/s3/.
2. En la barra de navegación de la parte superior de la página, elija el nombre de la región de AWS que aparece. A continuación, elija la región a la que desea cambiar.
3. En el panel de navegación izquierdo, elija Grupos de Storage Lens.
4. Elija Crear grupo de Lente de almacenamiento.
5. En General, vea la Región de origen e introduzca el nombre de su grupo de lentes de almacenamiento.
6. En Ámbito, elija el filtro que desea aplicar a su grupo de Lente de almacenamiento. Para aplicar varios filtros, elíjalos y, a continuación, elija el operador lógico AND u OR.

   - Para el filtro Prefijos, elija Prefijos e introduzca una cadena de prefijos. Para añadir varios prefijos, elija Agregar prefijo. Para eliminar un prefijo, elija Eliminar que está ubicado junto al prefijo que desea eliminar.
   - Para el filtro Etiquetas de objetos, elija Etiquetas de objeto e introduzca el par clave-valor del objeto. A continuación, elija Agregar etiqueta. Para eliminar una etiqueta, elija Eliminar junto a la etiqueta que desee eliminar.
   - Para el filtro Sufijos, elija Sufijos e introduzca una cadena de sufijos. Para añadir varios sufijos, elija Agregar sufijo. Para eliminar un sufijo, elija Eliminar que está ubicado junto al sufijo que desea eliminar.
   - Para el filtro Antigüedad, especifique el rango de antigüedad de objeto en días. Elija Especificar la edad mínima del objeto e introduzca la edad mínima del objeto. A continuación, elija Especificar la edad máxima del objeto e introduzca la edad máxima del objeto.
   - Para el filtro Tamaño, especifique el rango de tamaño de objeto y la unidad de medida. Elija Especificar el tamaño mínimo del objeto e introduzca el tamaño mínimo del objeto. Elija Especificar el tamaño máximo del objeto e introduzca el tamaño máximo del objeto.

7. (Opcional) Para las etiquetas de recursos de AWS, añada el par clave-valor y, a continuación, seleccione Agregar etiqueta.
8. Seleccione Crear grupo de Lente de almacenamiento.

Uso de la AWS CLI

El siguiente comando de AWS CLI de ejemplo crea un grupo de Storage Lens. Para utilizar este comando de ejemplo, sustituya *user input placeholders* por su propia información.

```
aws s3control create-storage-lens-group --account-id 111122223333 \
--region us-east-1 --storage-lens-group=file://./marketing-department.json
```

El siguiente comando de ejemplo de AWS CLI crea un grupo de Lente de almacenamiento con dos etiquetas de recursos de AWS. Para utilizar este comando de ejemplo, sustituya *user input placeholders* por su propia información.

```
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
```

Para ver configuraciones de JSON de ejemplo, consulte Configuración de grupos de Lente de almacenamiento.

Uso de AWS SDK para Java

El siguiente ejemplo de AWS SDK for Java crea un grupo de Lente de almacenamiento. Para utilizar este ejemplo, reemplace los *user input placeholders* con su propia información.

Example — Crear un grupo de Lente de almacenamiento con un solo filtro

El ejemplo siguiente crea un grupo de Storage Lens denominado *Marketing-Department*: Este grupo tiene un filtro de antigüedad de objetos que especifica el rango de antigüedad como de *90* a *30* días. Para utilizar este ejemplo, reemplace los *user input placeholders* con su propia información.

```
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 — Crear un grupo de Lente de almacenamiento con un operador **AND** que incluya varios filtros

El ejemplo siguiente crea un grupo de Storage Lens denominado *Marketing-Department*: Este grupo usa el operador AND para indicar que los objetos deben coincidir con todas las condiciones de filtro. Para utilizar este ejemplo, reemplace los *user input placeholders* con su propia información.

```
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 — Crear un grupo de Lente de almacenamiento con un operador **OR** que incluya varios filtros

El ejemplo siguiente crea un grupo de Storage Lens denominado *Marketing-Department* Este grupo usa un operador OR para aplicar un filtro de prefijo (*prefix-1*, *prefix-2*, *prefix3/sub-prefix-1*) o un filtro de tamaño de objeto con un rango de tamaño entre *1000* bytes y *6000* bytes. Para utilizar este ejemplo, reemplace los *user input placeholders* con su propia información.

```
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 — Crear un grupo de Lente de almacenamiento con un solo filtro y dos etiquetas de recursos de AWS

En el siguiente ejemplo, se crea un grupo de Lente de almacenamiento denominado *Marketing-Department* que tiene un filtro de sufijo. En este ejemplo también se agregan dos etiquetas de recursos AWS al grupo de Lente de almacenamiento. Para utilizar este ejemplo, reemplace los *user input placeholders* con su propia información.

```
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();
        }
    }
}
```

Para ver configuraciones de JSON de ejemplo, consulte Configuración de grupos de Lente de almacenamiento.

Asociación o eliminación de grupos de S3 Storage Lens a o desde su panel

Una vez que se haya actualizado al nivel avanzado en Amazon S3 Storage Lens, puede asociar un grupo de Storage Lens a su panel. Si tiene varios grupos de Storage Lens, puede incluir o excluir los grupos que desee.

Sus grupos de Storage Lens deben encontrarse en la región de origen designada en la cuenta del panel. Después de asociar un grupo de Storage Lens a su panel, recibirá los datos adicionales de agregación de grupos de Storage Lens en su exportación de métricas en un plazo de 48 horas.

> **Note**
>
> Si desea ver las métricas agregadas para su grupo de Storage Lens, debe asociarlas al panel de Storage Lens. Para ver ejemplos de los archivos de configuración JSON del grupo de Storage Lens, consulte Ejemplo de configuración de S3 Storage Lens con grupos de Storage Lens en JSON.

Asociación de un grupo de Storage Lens a un panel de S3 Storage Lens

Para asociar un grupo de Storage Lens a un panel de Storage Lens

1. Inicie sesión en la AWS Management Console y abra la consola de Amazon S3 en https://console.aws.amazon.com/s3/.
2. En el panel de navegación izquierdo, en Storage Lens, elija Paneles.
3. Seleccione el botón de opción del panel de Storage Lens al que desee asociar un grupo de Storage Lens.
4. Elija Editar.
5. En Metrics selection (Selección de métricas), elija Advanced metrics and recommendations (Métricas y recomendaciones avanzadas).

6. Seleccione la Agregación de grupos de Storage Lens.

> **Note**
>
> De forma predeterminada, también está seleccionada la opción Métricas avanzadas. Sin embargo, también puede anular la selección de esta configuración, ya que no es necesaria para agregar los datos de los grupos de Storage Lens.

7. Desplácese hasta Agregación de grupos de Storage Lens y especifique el grupo o los grupos de Storage Lens que desee incluir o excluir en la agregación de datos. Puede utilizar las siguientes opciones de filtrado:

   - Si desea incluir determinados grupos de Storage Lens, elija Incluir grupos de Storage Lens. En Grupos de Storage Lens para incluir, seleccione sus grupos de Storage Lens.
   - Si desea incluir todos los grupos de Storage Lens, seleccione Incluir todos los grupos de Storage Lens de la región de origen en esta cuenta.
   - Si desea excluir determinados grupos de Storage Lens, elija Excluir grupos de Storage Lens. En Grupos de Storage Lens para excluir, seleccione los grupos de Storage Lens que desee excluir.

8. Elija Guardar cambios. Si ha configurado los grupos de Storage Lens correctamente, verá los datos de agregación de grupo de Storage Lens adicionales en su panel en un plazo de 48 horas.

Eliminación de un grupo de Storage Lens de un panel de S3 Storage Lens

Para eliminar un grupo de Storage Lens de un panel de S3 Storage Lens

1. Inicie sesión en la AWS Management Console y abra la consola de Amazon S3 en https://console.aws.amazon.com/s3/.
2. En el panel de navegación izquierdo, en Storage Lens, elija Paneles.
3. Elija el botón de opción del panel de Storage Lens del que desee eliminar un grupo de Storage Lens.
4. Elija Ver la configuración del panel.
5. Elija Editar.
6. Desplácese hasta la sección Selección de métricas.
7. En la sección Agregación de grupos de Storage Lens, elija la X situada junto al grupo de Storage Lens que desee eliminar. De este modo, se elimina el grupo de Storage Lens.

Si ha incluido todos los grupos de Storage Lens en su panel, desactive la casilla situada junto a Incluir todos los grupos de Storage Lens de la región de origen en esta cuenta.

8. Elija Guardar cambios.

> **Note**
>
> El panel tardará hasta 48 horas en reflejar las actualizaciones de la configuración.

Visualización de los datos de grupos de Storage Lens

Puede visualizar sus datos de grupos de Storage Lens asociando el grupo al panel de Amazon S3 Storage Lens. Una vez que haya incluido el grupo de Storage Lens en la agregación de grupos de Storage Lens de la configuración del panel, pueden pasar hasta 48 horas para que los datos del grupo de Storage Lens se muestren en su panel.

Una vez actualizada la configuración del panel, todos los grupos de Storage Lens recién asociados aparecen en la lista de recursos disponibles en la pestaña Grupos de Storage Lens. También puede analizar aún más el uso del almacenamiento en la pestaña Información general dividiendo los datos por otra dimensión. Por ejemplo, puede elegir uno de los elementos que figuran en las 3 categorías principales y, a continuación, Analizar por para dividir los datos por otra dimensión. No puede aplicar la misma dimensión que el propio filtro.

> **Note**
>
> No puede aplicar un filtro de grupo de Storage Lens junto con un filtro de prefijo, ni a la inversa. Tampoco puede analizar aún más un grupo de Storage Lens mediante un filtro de prefijo.

Puede utilizar la pestaña Grupos de Storage Lens del panel de Amazon S3 Storage Lens para personalizar la visualización de los datos de los grupos de Storage Lens asociados a su panel. Puede visualizar los datos de algunos grupos de Storage Lens que están asociados a su panel, o todos ellos.

Al visualizar los datos de los grupos de Storage Lens en el panel de S3 Storage Lens, tenga en cuenta lo siguiente:

- S3 Storage Lens agrega las métricas de uso para un objeto en todos los grupos de Storage Lens coincidentes. Por lo tanto, si un objeto cumple las condiciones de filtrado de dos o más grupos de Storage Lens, verá recuentos repetidos para el mismo objeto a lo largo de su uso de almacenamiento.
- Los objetos deben coincidir con los filtros que incluya en sus grupos de Storage Lens. Si ningún objeto coincide con los filtros que incluye en su grupo de Storage Lens, no se generará ninguna métrica. Para determinar si hay objetos sin asignar, compruebe el recuento total de objetos en el panel en el nivel de cuenta y de bucket.

Actualización de un grupo de Storage Lens

En los siguientes ejemplos se muestra cómo actualizar un grupo de Amazon S3 Storage Lens. Puede actualizar un grupo de Storage Lens mediante la consola de Amazon S3, AWS Command Line Interface (AWS CLI) y AWS SDK for Java.

Uso de la consola de S3

Para actualizar un grupo de Storage Lens

1. Inicie sesión en la AWS Management Console y abra la consola de Amazon S3 en https://console.aws.amazon.com/s3/.
2. En el panel de navegación izquierdo, elija Grupos de Storage Lens.
3. En Grupos de Storage Lens, elija el grupo de Storage Lens que desee actualizar.
4. En Ámbito, elija Editar.
5. En la página Ámbito, seleccione el filtro que desee aplicar a su grupo de Storage Lens. Para aplicar varios filtros, selecciónelos y elija el operador lógico AND u OR.
   - Para el filtro Prefijos, seleccione Prefijos e introduzca una cadena de prefijos. Para añadir varios prefijos, elija Agrgar prefijo. Para eliminar un prefijo, elija Eliminar que está ubicado junto al prefijo que desea eliminar.
   - Para el filtro Etiquetas de objetos, introduzca el par clave-valor para el objeto. A continuación, elija Agregar etiqueta. Para eliminar una etiqueta, elija Eliminar junto a la etiqueta que desee eliminar.
   - Para el filtro Sufijos, seleccione Sufijos e introduzca una cadena de sufijos. Para añadir varios sufijos, elija Agregar sufijo. Para eliminar un sufijo, elija Eliminar que está ubicado junto al sufijo que desea eliminar.

- Para el filtro Antigüedad, especifique el rango de edad de objeto en días. Elija Especificar la edad mínima del objeto e introduzca la edad mínima del objeto. En Especificar la edad máxima del objeto, introduzca la edad máxima del objeto.
- Para el filtro Tamaño, especifique el rango de tamaño de objeto y la unidad de medida. Elija Especificar el tamaño mínimo del objeto e introduzca el tamaño mínimo del objeto. En Especificar el tamaño máximo del objeto, introduzca el tamaño máximo del objeto.

6. Elija Guardar cambios. Aparecerá la página de detalles del grupo de Storage Lens.
7. (Opcional) Si desea añadir una nueva etiqueta de recurso de AWS, desplácese hasta la sección de etiquetas de recursos de AWS y, a continuación, seleccione Agregar etiquetas. Aparece la página Add tags (Agregar etiquetas).

   Agregue el nuevo par clave-valor y, a continuación, elija Guardar cambios. Aparecerá la página de detalles del grupo de Storage Lens.
8. (Opcional) Si desea eliminar una etiqueta de recurso AWS, desplácese hasta la sección de etiquetas de recursos de AWS y seleccione la etiqueta de recurso. A continuación, elija Delete (Eliminar). Aparecerá el cuadro de diálogo Eliminar etiquetas de AWS.

   Vuelva a elegir Eliminar para eliminar permanentemente la etiqueta de recurso de AWS.

   > **Note**
   >
   > Después de eliminar permanentemente una etiqueta de recurso de AWS, no se puede restaurar.

Utilización de la AWS CLI

El siguiente comando de ejemplo de AWS CLI devuelve los detalles de configuración de un grupo de Storage Lens denominado *marketing-department*. Para utilizar este comando de ejemplo, sustituya *user input placeholders* por su propia información.

```
aws s3control get-storage-lens-group --account-id 111122223333 \
--region us-east-1 --name marketing-department
```

En el siguiente ejemplo de AWS CLI, se actualiza un grupo de Storage Lens. Para utilizar este comando de ejemplo, sustituya *user input placeholders* por su propia información.

```
aws s3control update-storage-lens-group --account-id 111122223333 \
--region us-east-1 --storage-lens-group=file://./marketing-department.json
```

Para ver configuraciones de JSON de ejemplo, consulte Configuración de grupos de Lente de almacenamiento.

Uso de AWS SDK para Java

El siguiente ejemplo de AWS SDK for Java devuelve los detalles de configuración del grupo de Storage Lens *Marketing-Department* de la cuenta *111122223333*. Para utilizar este ejemplo, reemplace los *user input placeholders* con su propia información.

```
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();
        }
    }
}
```

En el siguiente ejemplo, se actualiza el grupo de Storage Lens denominado *Marketing-Department* en la cuenta *111122223333*. En este ejemplo, se actualiza el ámbito del panel para incluir objetos que coincidan con alguno de los siguientes sufijos: *.png*, *.gif*, *.jpg* o *.jpeg*. Para utilizar este ejemplo, reemplace los *user input placeholders* con su propia información.

```
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();
        }
    }
}
```

Para ver configuraciones de JSON de ejemplo, consulte Configuración de grupos de Lente de almacenamiento.

Administración de etiquetas de recursos de AWS con grupos de Storage Lens

Cada grupo de Amazon S3 Storage Lens se cuenta como un recurso de AWS con su propio Nombre de recurso de Amazon (ARN). Por lo tanto, al configurar su grupo de Storage Lens, si lo desea, puede añadir etiquetas de recursos de AWS al grupo. Puede añadir un máximo de 50 etiquetas a cada grupo de Storage Lens. Para crear un grupo de Storage Lens con etiquetas, debe tener los permisos `s3:CreateStorageLensGroup` y `s3:TagResource`.

Puede usar etiquetas de recursos de AWS para clasificar los recursos por departamento, línea de negocio o proyecto. Esto resulta útil cuando se tienen muchos recursos del mismo tipo. Al aplicar etiquetas, puede identificar rápidamente un grupo de Storage Lens específico en función de las etiquetas que le haya asignado. También puede utilizar etiquetas para realizar un seguimiento de los costes y asignarlos.

Además, cuando agrega una etiqueta de recurso de AWS a su grupo de Storage Lens, activa el control de acceso basado en atributos (ABAC). ABAC es una estrategia de autorización que define permisos basados en atributos, en este caso, etiquetas. También puede utilizar las condiciones que

especifican etiquetas de recursos en sus políticas de IAM para controlar el acceso a los recursos de AWS.

Puede editar las claves y los valores de las etiquetas y también puede eliminar etiquetas de un recurso en cualquier momento. Además, debe tener en cuenta las siguientes limitaciones:

- Las claves y los valores de las etiquetas distinguen entre mayúsculas y minúsculas.
- Si añade una etiqueta con la misma clave que una etiqueta existente en ese recurso, el nuevo valor sobrescribirá al antiguo.
- Si elimina un recurso, también se eliminará cualquier etiqueta asignada a dicho recurso.
- No incluya datos privados o confidenciales en sus etiquetas de recursos de AWS.
- No se admiten etiquetas del sistema (con claves de etiqueta que comiencen por `aws:`).
- La longitud de cada clave de etiqueta no puede superar los 128 caracteres. La longitud del valor de cada etiqueta no puede superar los 256 caracteres.

En los siguientes ejemplos se muestra cómo utilizar etiquetas de recursos de AWS con grupos de Storage Lens.

Temas

- Adición de una etiqueta de recurso de AWS a un grupo de Storage Lens
- Actualización de los valores de las etiquetas de grupo de Storage Lens
- Eliminación de una etiqueta de recurso de AWS de un grupo de Storage Lens
- Enumeración de etiquetas de grupo de Storage Lens

Adición de una etiqueta de recurso de AWS a un grupo de Storage Lens

En los siguientes ejemplos se muestra cómo agregar etiquetas de recursos de AWS a un grupo de Amazon S3 Storage Lens. Puede añadir etiquetas de recursos mediante la consola de Amazon S3, AWS Command Line Interface (AWS CLI) y AWS SDK for Java.

Uso de la consola de S3

Para agregar una etiqueta de recurso de AWS a un grupo de Storage Lens

1. Inicie sesión en la AWS Management Console y abra la consola de Amazon S3 en https://console.aws.amazon.com/s3/.
2. En el panel de navegación izquierdo, elija Grupos de Storage Lens.

3. En los Grupos de Storage Lens, elija el grupo de Storage Lens que desee actualizar.
4. En Etiquetas de recursos de AWS, elija Agregar etiquetas.
5. En la página Agregar etiquetas, añada el nuevo par clave-valor.

> **Note**
>
> Si añade una nueva etiqueta con la misma clave que una etiqueta existente, se sobrescribirá el valor de la etiqueta anterior.

6. (Opcional) Para añadir más de una etiqueta nueva, vuelve a elegir Agregar etiqueta para seguir añadiendo nuevas entradas. Puede añadir un máximo de 50 etiquetas de recursos de AWS a su grupo de Storage Lens.
7. (Opcional) Si desea eliminar una entrada recién agregada, elija Eliminar junto a la etiqueta que desea eliminar.
8. Elija Save changes (Guardar cambios).

Utilización de la AWS CLI

El siguiente comando de ejemplo de AWS CLI añade dos etiquetas de recursos a un grupo de Storage Lens existente denominado *marketing-department*. Para utilizar este comando de ejemplo, sustituya *user input placeholders* por su propia información.

```
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
```

Uso de AWS SDK para Java

El siguiente ejemplo de AWS SDK for Java añade dos etiquetas de recursos de AWS a un grupo de Storage Lens existente. Para utilizar este ejemplo, reemplace los *user input placeholders* con su propia información.

```
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();
        }
    }
}
```

Actualización de los valores de las etiquetas de grupo de Storage Lens

En los siguientes ejemplos se muestra cómo actualizar los valores de las etiquetas de grupo de Storage Lens mediante la consola de Amazon S3, AWS Command Line Interface (AWS CLI) y AWS SDK for Java.

Uso de la consola de S3

Para actualizar una etiqueta de recursos de AWS para un grupo de Storage Lens

1. Inicie sesión en la AWS Management Console y abra la consola de Amazon S3 en https://console.aws.amazon.com/s3/.
2. En el panel de navegación izquierdo, elija Grupos de Storage Lens.
3. En Grupos de Storage Lens, elija el grupo de Storage Lens que desee actualizar.
4. En las Etiquetas de recursos de AWS, selecciona la etiqueta que desee actualizar.
5. Añada el nuevo valor de etiqueta usando la misma clave del par clave-valor que desee actualizar. Haga clic en el icono de marca de verificación para actualizar el valor de la etiqueta.

   > **Note**
   >
   > Si añade una nueva etiqueta con la misma clave que una etiqueta existente, se sobrescribirá el valor de la etiqueta anterior.

6. (Opcional) Si desea añadir nuevas etiquetas, elija Agregar etiqueta para añadir nuevas entradas. Aparece la página Add tags (Agregar etiquetas).

   Puede agregar hasta 50 etiquetas de recursos de AWS para su grupo de Storage Lens. Cuando haya terminado de añadir etiquetas, elija Guardar cambios.
7. (Opcional) Si desea eliminar una entrada recién agregada, elija Eliminar junto a la etiqueta que desea eliminar. Cuando haya terminado de eliminar etiquetas, elija Guardar cambios.

Utilización de la AWS CLI

El siguiente comando AWS CLI de ejemplo actualiza dos valores de etiqueta para el grupo de Storage Lens denominado *marketing-department*. Para utilizar este comando de ejemplo, sustituya *user input placeholders* por su propia información.

```
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
```

Uso de AWS SDK para Java

El siguiente ejemplo de AWS SDK for Java actualiza dos valores de etiqueta de grupo de Storage Lens. Para utilizar este ejemplo, reemplace los *user input placeholders* con su propia información.

```
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();
        }
    }
}
```

Eliminación de una etiqueta de recurso de AWS de un grupo de Storage Lens

En los siguientes ejemplos se muestra cómo eliminar una etiqueta de recurso de AWS de un grupo de Storage Lens. Puede eliminar etiquetas mediante la consola de Amazon S3, AWS Command Line Interface (AWS CLI) y AWS SDK for Java.

Uso de la consola de S3

Para eliminar una etiqueta de recurso de AWS de un grupo de Storage Lens

1. Inicie sesión en la AWS Management Console y abra la consola de Amazon S3 en https://console.aws.amazon.com/s3/.
2. En el panel de navegación izquierdo, elija Grupos de Storage Lens.
3. En Grupos de Storage Lens, elija el grupo de Storage Lens que desea actualizar.
4. En Etiquetas de recursos de AWS, seleccione el par clave-valor que desea eliminar.
5. Elija Eliminar. Aparece el cuadro de diálogo Eliminar etiquetas de recursos de AWS.

> **Note**
>
> Si se utilizan etiquetas para controlar el acceso, continuar con esta acción puede afectar a los recursos relacionados. Después de eliminar una etiqueta de forma permanente, no se puede restaurar.

6. Elija Eliminar para eliminar el par clave-valor de forma permanente.

Utilización de la AWS CLI

El siguiente comando de AWS CLI elimina dos etiquetas de recursos de AWS de un grupo de Storage Lens existente: para usar este comando de ejemplo, reemplace los *user input placeholders* por su propia información.

```
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
```

Uso de AWS SDK para Java

El siguiente ejemplo de AWS SDK for Java elimina dos etiquetas de recursos de AWS del Nombre de Recurso de Amazon (ARN) del grupo de Storage Lens que especifica en la cuenta *111122223333*. Para utilizar este ejemplo, reemplace los *user input placeholders* con su propia información.

```
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();
        }
    }
}
```

Enumeración de etiquetas de grupo de Storage Lens

En los siguientes ejemplos se muestra cómo enumerar las etiquetas de recursos de AWS asociadas a un grupo de Storage Lens. Puede enumerar etiquetas mediante la consola de Amazon S3, AWS Command Line Interface (AWS CLI) y AWS SDK for Java.

Uso de la consola de S3

Para revisar la lista de etiquetas y los valores de las etiquetas de un grupo de Storage Lens

1. Inicie sesión en la AWS Management Console y abra la consola de Amazon S3 en https://console.aws.amazon.com/s3/.
2. En el panel de navegación izquierdo, elija Grupos de Storage Lens.
3. En Grupos de Storage Lens, elija el grupo de Storage Lens que le interesa.
4. Desplácese hasta la sección Etiquetas de recursos de AWS. Todas las etiquetas de recursos de AWS definidas por el usuario que se añaden a su grupo de Storage Lens aparecen junto con sus valores de etiqueta.

Utilización de la AWS CLI

El siguiente comando de ejemplo de AWS CLI muestra todos los valores de etiqueta de grupo de Storage Lens del grupo Storage Lens denominado *marketing-department*. Para utilizar este comando de ejemplo, sustituya *user input placeholders* por su propia información.

```
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
```

Uso de AWS SDK para Java

En el siguiente ejemplo de AWS SDK for Java se muestran los valores de etiqueta de grupo de Storage Lens para el Nombre de recurso de Amazon (ARN) del grupo Storage Lens que especifique. Para utilizar este ejemplo, reemplace los *user input placeholders* con su propia información.

```
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();
        }
    }
}
```

Enumeración de todos los grupos de Storage Lens

En los ejemplos siguientes, se muestra cómo enumerar todos los grupos de Amazon S3 Storage Lens de una Cuenta de AWS y una región de origen. En estos ejemplos se muestra cómo enumerar todos los grupos de Storage Lens mediante la consola de Amazon S3, AWS Command Line Interface (AWS CLI) yAWS SDK for Java.

Uso de la consola de S3

Para enumerar todos los grupos de Storage Lens en una cuenta y región de origen

1. Inicie sesión en la AWS Management Console y abra la consola de Amazon S3 en https://console.aws.amazon.com/s3/.
2. En el panel de navegación izquierdo, elija Grupos de Storage Lens.
3. En Grupos de Storage Lens, se muestra la lista de grupos de Storage Lens de su cuenta.

Utilización de la AWS CLI

El siguiente ejemplo de AWS CLI muestra todos los grupos de Storage Lens para su cuenta. Para utilizar este comando de ejemplo, sustituya *user input placeholders* por su propia información.

```
aws s3control list-storage-lens-groups --account-id 111122223333 \
--region us-east-1
```

Uso de AWS SDK para Java

El siguiente ejemplo de AWS SDK for Java muestra los grupos de Storage Lens para la cuenta *111122223333*. Para utilizar este ejemplo, reemplace los *user input placeholders* con su propia información.

```
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();
        }
    }
}
```

Visualización de los detalles del grupo de Storage Lens

En los siguientes ejemplos se muestra cómo ver los detalles de configuración del grupo de Amazon S3 Storage Lens. Puede ver estos detalles mediante la consola de Amazon S3, AWS Command Line Interface (AWS CLI) y AWS SDK for Java.

Uso de la consola de S3

Para ver los detalles de configuración del grupo de Storage Lens

1. Inicie sesión en la AWS Management Console y abra la consola de Amazon S3 en https://console.aws.amazon.com/s3/.
2. En el panel de navegación izquierdo, elija Grupos de Storage Lens.
3. En los Grupos de Storage Lens, elija el botón de opción situado junto al grupo de Storage Lens que le interese.
4. Elija View details (Ver detalles). Ahora puede revisar los detalles de su grupo de Storage Lens.

Utilización de la AWS CLI

El siguiente ejemplo de AWS CLI devuelve los detalles de configuración de un grupo de Storage Lens. Para utilizar este comando de ejemplo, sustituya *user input placeholders* por su propia información.

```
aws s3control get-storage-lens-group --account-id 111122223333 \
--region us-east-1 --name marketing-department
```

Uso de AWS SDK para Java

El siguiente ejemplo de AWS SDK for Java devuelve los detalles de configuración del grupo de Storage Lens denominado *Marketing-Department* en la cuenta *111122223333*. Para utilizar este ejemplo, reemplace los *user input placeholders* con su propia información.

```
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();
        }
    }
}
```

Eliminación de un grupo de Storage Lens

En los siguientes ejemplos se muestra cómo eliminar un grupo de Amazon S3 Storage Lens mediante la consola de Amazon S3, AWS Command Line Interface (AWS CLI) yAWS SDK for Java.

Uso de la consola de S3

Para eliminar un grupo de Storage Lens

1. Inicie sesión en la AWS Management Console y abra la consola de Amazon S3 en https://console.aws.amazon.com/s3/.
2. En el panel de navegación izquierdo, elija Grupos de Storage Lens.
3. En Grupos de Storage Lens, elija el botón de opción situado junto al grupo de Storage Lens que desea eliminar.
4. Elija Eliminar. Aparece un cuadro de diálogo Eliminar grupo de Storage Lens.
5. Vuelva a elegir Eliminar para eliminar permanentemente su grupo de Storage Lens.

> **Note**
>
> Después de eliminar un grupo de Storage Lens, no se puede restaurar.

Utilización de la AWS CLI

El siguiente ejemplo de AWS CLI elimina el grupo de Storage Lens denominado *marketing-department*. Para utilizar este comando de ejemplo, sustituya *user input placeholders* por su propia información.

```
aws s3control delete-storage-lens-group --account-id 111122223333 \
--region us-east-1 --name marketing-department
```

Uso de AWS SDK para Java

El siguiente ejemplo de AWS SDK for Java elimina el grupo de Storage Lens denominado *Marketing-Department* en la cuenta *111122223333*. Para utilizar este ejemplo, reemplace los *user input placeholders* con su propia información.

```
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();
        }
    }
}
```

Seguimiento de solicitudes de Amazon S3 mediante AWS X-Ray

AWS X-Ray recopila datos sobre las solicitudes que su aplicación procesa. Usted luego puede ver y filtrar los datos para identificar y solucionar problemas y errores de rendimiento en las aplicaciones distribuidas y en la arquitectura de microservicios. Para cada solicitud rastreada que se envía a la aplicación, se muestra información detallada sobre la solicitud, la respuesta y las llamadas que la aplicación realiza a los recursos de AWS, los microservicios, las bases de datos y las API web de HTTP.

Para obtener más información, consulte ¿Qué es AWS X-Ray? en la Guía para desarrolladores de AWS X-Ray.

Temas

- Cómo funciona X-Ray con Amazon S3
- Regiones disponibles

Cómo funciona X-Ray con Amazon S3

AWS X-Ray admite la propagación de contexto de seguimiento para Amazon S3, por lo que puede ver las solicitudes de extremo a extremo cuando se desplazan por toda la aplicación. X-Ray agrega los datos que generan los servicios individuales, como Amazon S3, AWS Lambda y Amazon EC2, y los numerosos recursos que conforman la aplicación. Le proporciona una visión general del rendimiento de la aplicación.

Amazon S3 se integra con X-Ray para propagar el contexto de seguimiento y ofrecer una cadena de solicitudes con nodos upstream y downstream . Si un servicio upstream incluye un encabezado de rastreo con formato válido con su solicitud S3, Amazon S3 pasa el encabezado de rastreo cuando entrega notificaciones de eventos a servicios downstream como Lambda, Amazon SQS y Amazon SNS. Si tiene todos estos servicios integrados activamente con X-Ray, están enlazados en una cadena de solicitudes para proporcionarle los detalles completos de sus solicitudes de Amazon S3.

Para enviar encabezados de rastreo de X-Ray mediante Amazon S3, debe incluir un X-Amzn-Trace-Id con formato en sus solicitudes. También puede instrumentar el cliente de Amazon S3 mediante los SDK de AWS X-Ray. Para obtener una lista de los SDK admitidos, consulte la documentación de AWS X-Ray.

Mapas de servicio

Los mapas de servicio de X-Ray muestran casi en tiempo real las relaciones entre Amazon S3 y otros servicios y recursos de AWS en la aplicación. Para ver las solicitudes de extremo a extremo mediante los mapas de servicio X-Ray, puede utilizar la consola de X-Ray para ver un mapa de las conexiones entre Amazon S3 y otros servicios que utiliza su aplicación. Puede detectar fácilmente dónde se producen altas latencias, visualizar la distribución de los nodos para estos servicios y, tras ello, desglosar hasta los servicios y rutas específicos que impactan en el rendimiento de la aplicación.

Análisis de X-Ray

También puede utilizar la consola de análisis de X-Ray para analizar rastros, ver métricas, como la latencia y las tasas de errores, y generar información para facilitar la identificación y la solución de problemas. Esta consola también muestra métricas tales como la latencia media y las tasas de error. Para obtener más información, consulte Consola de AWS X-Ray en la Guía para desarrolladores de AWS X-Ray.

Regiones disponibles

El soporte de AWS X-Ray para Amazon S3 está disponible en todas las regiones de AWS X-Ray. Para obtener más información, consulte Amazon S3 y AWS X-Ray en la Guía para desarrolladores de AWS X-Ray.

Alojamiento de un sitio web estático mediante Amazon S3

Puede utilizar Amazon S3 para alojar un sitio web estático. En un sitio web estático, cada página web incluye contenido estático. También pueden contener scripts del lado del cliente.

Por el contrario, un sitio web dinámico depende del procesamiento en el lado del servidor, incluidos los scripts del lado del servidor, como en PHP, JSP o ASP.NET. Amazon S3 no admite el scripting del lado del servidor, pero AWS dispone de otros recursos para alojar sitios web dinámicos. Para obtener más información sobre el alojamiento de sitios web en AWS, consulte Alojamiento web.

> **Note**
>
> Puede utilizar la consola de AWS Amplify para alojar una aplicación web de una sola página. La consola de AWS Amplify admite aplicaciones de una sola página creadas con marcos de aplicaciones de una sola página (por ejemplo: React JS, Vue JS, Angular JS y Nuxt) y generadores de sitios estáticos (por ejemplo: Gatsby JS, React-static, Jekyll y Hugo). Para obtener más información, consulte Introducción en la Guía del usuario de la consola de AWS Amplify.
>
> Los puntos de enlace de sitio web de Amazon S3 no admiten HTTPS. Si desea usar HTTPS, puede emplear Amazon CloudFront para atender a un sitio web estático alojado en Amazon S3. Para obtener más información, consulte ¿Cómo uso CloudFront para que atienda solicitudes HTTPS de mi bucket de Amazon S3? Para utilizar HTTPS con un dominio personalizado, consulte Configuración de un sitio web estático mediante un dominio personalizado registrado con Route 53.

Para obtener más información sobre cómo alojar un sitio web estático en Simple Storage Service (Amazon S3), incluidas las instrucciones y las explicaciones paso a paso, consulte los siguientes temas:

Temas

- Puntos de enlace de sitio web
- Habilitar el alojamiento de sitios web
- Configurar un documento de índice
- Configurar un documento de error personalizado
- Configurar permisos para el acceso a sitios web

- (Opcional) Registro del tráfico web
- (Opcional) Configuración del redireccionamiento de páginas web
- Uso compartido de recursos entre orígenes (CORS)

Puntos de enlace de sitio web

Cuando configura el bucket como un sitio web estático, el sitio web está disponible en el punto de enlace de sitio web de la Región de AWS específica del bucket. Los puntos de enlace de sitio web son distintos de los puntos de enlace a donde envía las solicitudes de la Application Programming Interface (API, Interfaz de programación de aplicaciones) de REST. Para obtener más información acerca de las diferencias entre los puntos de enlace, consulte Diferencias clave entre el punto de enlace de un sitio web y un punto de enlace de la API de REST.

En función de la región, el punto de enlace del sitio web de Amazon S3 siguen uno de estos dos formatos.

- s3-website guion (-) región - `http://bucket-name.s3-website-Region.amazonaws.com`
- s3-web punto (.) Región - `http://bucket-name.s3-website.Region.amazonaws.com`

Estas URL devuelven el documento de índice predeterminado que configuró para el sitio web. Para obtener una lista completa de los puntos de enlace del sitio web de Amazon S3, consulte Puntos de enlace de sitio web de Amazon S3.

> **Note**
>
> Para aumentar la seguridad de los sitios web estáticos de Amazon S3, los dominios de punto de conexión del sitio web de Amazon S3 (por ejemplo, s3-website-us-east-1.amazonaws.com o s3-website.ap-south-1.amazonaws.com) se registran en la lista de sufijos públicos (PSL). Para mayor seguridad, se recomienda que utilice cookies con un prefijo `__Host-` en caso de que necesite configurar cookies confidenciales en el nombre de dominio de sus sitios web estáticos de Amazon S3. Esta práctica le ayudará a proteger su dominio de los intentos de falsificación de solicitudes entre sitios (CSRF). Para obtener más información, consulte la página de configuración de cookies en la red de desarrolladores de Mozilla.

Si desea que su sitio web sea público, debe hacer que el contenido sea legible públicamente para que los clientes puedan acceder a él en el punto de conexión del sitio web. Para obtener más información, consulte Configurar permisos para el acceso a sitios web.

> **Important**
>
> Los puntos de enlace del sitio web de Amazon S3 no admiten HTTPS ni puntos de acceso. Si desea usar HTTPS, puede emplear Amazon CloudFront para atender a un sitio web estático alojado en Amazon S3. Para obtener más información, consulte ¿Cómo uso CloudFront para que atienda solicitudes HTTPS de mi bucket de Amazon S3? Para utilizar HTTPS con un dominio personalizado, consulte Configuración de un sitio web estático mediante un dominio personalizado registrado con Route 53.
>
> Los buckets de pago por solicitante no permiten el acceso mediante puntos de enlace de sitio web. Cualquier solicitud a un bucket de este tipo recibe una respuesta 403 Acceso denegado . Para obtener más información, consulte Utilización de buckets de pago por solicitante para transferencias de almacenamiento y uso.

Temas

Ejemplos de puntos de enlace de sitio web

Los siguientes ejemplos muestran cómo acceder a un bucket de Amazon S3 que está configurado como sitio web estático.

Example — Solicitud de un objeto en el nivel raíz

Para solicitar un objeto específico que esté almacenado en el nivel raíz del bucket, utilice la siguiente estructura de URL.

```
http://bucket-name.s3-website.Region.amazonaws.com/object-name
```

Por ejemplo, la siguiente dirección URL solicita el objeto `photo.jpg` que está almacenado en el nivel raíz del bucket.

```
http://example-bucket.s3-website.us-west-2.amazonaws.com/photo.jpg
```

Example — Solicitud de un objeto en un prefijo

Para solicitar un objeto almacenado en una carpeta del bucket, utilice esta estructura de URL.

```
http://bucket-name.s3-website.Region.amazonaws.com/folder-name/object-name
```

La siguiente dirección URL solicita el objeto `docs/doc1.html` en el bucket.

```
http://example-bucket.s3-website.us-west-2.amazonaws.com/docs/doc1.html
```

Agregar un CNAME DNS

Si tiene un dominio registrado, puede añadir una entrada CNAME de DNS para asociar el punto de enlace del sitio web de Amazon S3. Por ejemplo, si ha registrado el dominio `www.example-bucket.com`, puede crear un bucket `www.example-bucket.com` y añadir un registro CNAME de DNS que se asocie a `www.example-bucket.com.s3-website.Region.amazonaws.com`. Todas las solicitudes a `http://www.example-bucket.com` serán direccionadas a `www.example-bucket.com.s3-website.Region.amazonaws.com`.

Para obtener más información, consulte Personalización de URL de Amazon S3 con registros CNAME.

Uso de un dominio personalizado con Route 53

En lugar de acceder al sitio web mediante un sitio web de punto de enlace de Amazon S3, puede utilizar su propio dominio registrado en Amazon Route 53 para servir su contenido, por ejemplo, `example.com`. Puede usar Amazon S3 con Route 53 para alojar un sitio web en el dominio raíz. Por ejemplo, si tiene el dominio raíz `example.com` y aloja su sitio web en Amazon S3, los visitantes de su sitio web pueden acceder al sitio desde su navegador entrando en `http://www.example.com` o `http://example.com`.

Para ver un tutorial de ejemplo, consulte Tutorial: Configuración de un sitio web estático mediante un dominio personalizado registrado con Route 53.

Diferencias clave entre el punto de enlace de un sitio web y un punto de enlace de la API de REST

Un punto de enlace de sitio web de Amazon S3 está optimizado para obtener acceso desde un navegador web. En la siguiente tabla se resumen las diferencias clave entre un punto de enlace de API de REST y un punto de enlace de sitio web.

Diferencia de la clave	Punto de enlace de la API de REST	Punto de enlace de sitio web
Control de acceso	Admite contenido público y privado	Admite solo contenido público que se puede leer
Gestión de mensaje de error	Devuelve una respuesta de error con formato XML	Devuelve un documento HTML
Compatibilidad de redirecci onamiento	No aplicable	Admite el redireccionamiento en el nivel de objeto y de bucket
Solicitudes admitidas	Admite todas las operaciones de bucket y objeto.	Solamente admite solicitudes GET y HEAD en los objetos
Respuestas a las solicitudes GET y HEAD en la raíz de un bucket	Devuelve una lista de todas las claves de objetos en el bucket	Devuelve un documento de índice que se especificó en la configuración del sitio web
Compatibilidad con la Secure Sockets Layer (SSL, Capa de conexión segura)	Admite conexiones SSL	No admite conexiones SSL

Para obtener una lista completa de los puntos de conexión de Amazon S3, consulte Puntos de conexión y cuotas de Amazon S3 en la Referencia general de AWS.

Habilitar el alojamiento de sitios web

Cuando configura un bucket como sitio web estático, debe habilitar el alojamiento de sitios web estáticos, configurar un documento de índice y establecer permisos.

Puede habilitar el alojamiento de sitios web estáticos mediante la consola de Amazon S3, la API de REST, los SDK de AWS, la AWS CLI o AWS CloudFormation.

Para configurar el sitio web con un dominio personalizado, consulte Tutorial: Configuración de un sitio web estático mediante un dominio personalizado registrado con Route 53.

Uso de la consola de S3

Para habilitar el alojamiento estático de sitios web

1. Inicie sesión en la AWS Management Console y abra la consola de Amazon S3 en https://console.aws.amazon.com/s3/.
2. En la lista Buckets, seleccione el nombre del bucket para el que desea habilitar el alojamiento de sitios web estáticos.
3. Seleccione Properties (Propiedades).
4. Elija Static website hosting (Alojamiento de sitios web estáticos), elija Edit (Editar).
5. Elija Use this bucket to host a website (Usar este bucket para alojar un sitio web).
6. En Static website hosting (Alojamiento de sitios web estáticos), elija Enable (Habilitar).
7. En Index Document (Documento de índice), escriba el nombre de archivo del documento de índice, normalmente `index.html`.

   El nombre del documento de índice distingue entre mayúsculas y minúsculas y debe coincidir exactamente con el nombre del archivo del documento de índice HTML que tiene previsto cargar en el bucket de S3. Al configurar un bucket para el alojamiento de sitios web, debe especificar un documento de índice. Amazon S3 devuelve este documento de índice cuando se reciben solicitudes en el dominio raíz o en cualquiera de las subcarpetas. Para obtener más información, consulte Configurar un documento de índice.
8. Si desea proporcionar su propio documento de error personalizado para los errores de clase 4XX, escriba el nombre de archivo del documento de error personalizado en Error document (Documento de error).

   El nombre del documento de error distingue entre mayúsculas y minúsculas y debe coincidir exactamente con el nombre del archivo del documento de error HTML que tiene previsto cargar

en el bucket de S3. Si no especifica un documento de error personalizado y se produce un error, Amazon S3 devuelve un documento de error HTML predeterminado. Para obtener más información, consulte Configurar un documento de error personalizado.

9. (Opcional) Si desea especificar reglas de redireccionamiento avanzadas, en Redirection rules (Reglas de redireccionamiento), especifique JSON para describir las reglas.

   Por ejemplo, puede dirigir condicionalmente las solicitudes según nombres de clave de objeto o prefijos específicos en la solicitud. Para obtener más información, consulte Configurar reglas de redireccionamiento para utilizar redireccionamiento condicional avanzado.

10. Elija Save changes (Guardar cambios).

    Amazon S3 permite activar el alojamiento de sitios web estáticos para su bucket. En la parte inferior de la página, en Static website hosting (Alojamiento de sitios web estáticos), verá el punto de conexión del sitio web para su bucket.

11. En Static website hosting (Alojamiento de sitios web estáticos), anote el valor de Endpoint (Punto de enlace).

    Endpoint (Punto de enlace) es el punto de conexión del sitio web de Amazon S3 para el bucket. Cuando termine de configurar el bucket como un sitio web estático, puede utilizar este punto de enlace para probar el sitio web.

Uso de la API de REST

Para obtener más información sobre cómo enviar solicitudes REST directamente para habilitar el alojamiento de sitios web estáticos, consulte las siguientes secciones en la Referencia de la API de Amazon Simple Storage Service:

- PUT Bucket website
- GET Bucket website
- DELETE Bucket website

Uso de los AWS SDK

Para alojar un sitio web estático en Amazon S3, debe configurar un bucket de S3 para el alojamiento de sitio web y cargar el contenido del sitio web en el bucket. También puede utilizar los AWS SDK para crear, actualizar y eliminar la configuración del sitio web mediante programación. Los SDK

proporcionan clases de encapsulamiento en toda la API de REST de Amazon S3. Si su aplicación lo requiere, puede enviar solicitudes de la API de REST directamente desde su aplicación.

.NET

El siguiente ejemplo muestra cómo usar AWS SDK for .NET para administrar la configuración de un sitio web para un bucket. Para agregar una configuración de sitio web a un bucket, proporcione el nombre del bucket y una configuración de sitio web. La configuración de sitio web debe incluir un documento de índice y puede contener un documento de error opcional. Estos documentos ya deben estar almacenados en el bucket. Para obtener más información, consulte la sección sobre Sitio web PUT Bucket. Para obtener más información acerca de la característica de sitio web de Amazon S3, consulte Alojamiento de un sitio web estático mediante Amazon S3.

El siguiente ejemplo de código C# agrega una configuración de sitio web al bucket específico. La configuración especifica los nombres tanto del documento de índice como el de error. Para obtener información acerca de cómo configurar y ejecutar ejemplos de código, consulte Introducción al SDK de AWS para .NET en la Guía para desarrolladores del SDK de AWS para .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

El siguiente ejemplo de PHP agrega una configuración de sitio web al bucket específico. El método `create_website_config` proporciona explícitamente los nombres de los documentos de error e índice. El ejemplo recupera también la configuración del sitio web e imprime la respuesta. Para obtener más información acerca de la característica de sitio web de Amazon S3, consulte Alojamiento de un sitio web estático mediante Amazon S3.

Para obtener más información acerca de la API del SDK de AWS para Ruby, consulte SDK de AWS para Ruby, versión 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
]);
```

Mediante AWS CLI

Para obtener más información acerca de cómo usar la AWS CLI para configurar un bucket de S3 como un sitio web estático, consulte website en la Referencia de comandos de la AWS CLI.

A continuación, debe configurar el documento de índice y establecer permisos. Para obtener más información, consulte Configurar un documento de índice y Configurar permisos para el acceso a sitios web.

También puede configurar opcionalmente un documento de error, un registro de tráfico web o una redirección.

Configurar un documento de índice

Cuando habilita el alojamiento de sitios web, también debe configurar y cargar un documento de índice. Un documento de índice es una página web que devuelve Amazon S3 cuando se realiza una solicitud a la raíz de un sitio web o cualquier subcarpeta. Por ejemplo, si un usuario introduce `http://www.example.com` en el navegador, el usuario no solicita ninguna página específica. En ese caso, Amazon S3 ofrece el documento de índice, al que a veces se denomina la página predeterminada.

Cuando habilite el alojamiento de sitio web estático para su bucket, escriba el nombre del documento de índice (por ejemplo: `index.html`). Después de habilitar el alojamiento de sitio web estático para el bucket, cargue un archivo HTML con el nombre del documento de índice en el bucket.

La barra diagonal en el URL raíz es opcional. Por ejemplo, si configura el sitio web con `index.html` como documento de índice, las siguientes URL devuelven `index.html`.

```
http://example-bucket.s3-website.Region.amazonaws.com/
http://example-bucket.s3-website.Region.amazonaws.com
```

Para obtener más información acerca de los puntos de enlace de sitio web de Amazon S3, consulte Puntos de enlace de sitio web.

Documento de índice y carpetas

En Amazon S3, un bucket es un contenedor plano de objetos. No proporciona ninguna organización jerárquica como hace el sistema de archivos en su equipo. Sin embargo, puede crear una jerarquía lógica al usar los nombres de clave de objeto que implican una estructura de carpeta.

Por ejemplo, tomemos el caso de un bucket con tres objetos y los siguientes nombres de clave. Aunque están almacenados sin una organización jerárquica, puede inferir la siguiente estructura lógica de carpeta de los nombres de clave.

- `sample1.jpg` El objeto es la raíz del bucket.
- `photos/2006/Jan/sample2.jpg` El objeto se encuentra en la subcarpeta `photos/2006/Jan`.
- `photos/2006/Feb/sample3.jpg` El objeto se encuentra en la subcarpeta `photos/2006/Feb`.

En la consola de Amazon S3, también puede crear una carpeta en un bucket. Por ejemplo, puede crear una carpeta denominada `photos`. Puede cargar objetos en el bucket o en la carpeta `photos` dentro del bucket. Si añade el objeto `sample.jpg` al bucket, el nombre de clave será `sample.jpg`. Si carga el objeto a la carpeta `photos`, el nombre de clave del objeto será `photos/sample.jpg`.

Si crea esa estructura de carpeta en el bucket, debe tener un documento de índice en cada nivel. En cada carpeta, el documento de índice debe tener el mismo nombre, por ejemplo, `index.html`. Cuando un usuario especifica un URL que es similar a la búsqueda de una carpeta, la presencia o ausencia de una barra diagonal determina el comportamiento del sitio web. Por ejemplo, el siguiente URL, con una barra diagonal, devuelve el documento de índice `photos/index.html`.

```
http://bucket-name.s3-website.Region.amazonaws.com/photos/
```

Sin embargo, si excluye la barra diagonal del URL anterior, Amazon S3 primero buscará un objeto `photos` en el bucket. Si no encuentra el objeto `photos`, busca un documento de índice, `photos/index.html`. Se encuentra el documento, Amazon S3 devuelve un mensaje `302 Found` e indica la clave `photos/`. Para las solicitudes posteriores a `photos/`, Amazon S3 devuelve `photos/index.html`. Si no encuentra el documento de índice, Amazon S3 devuelve un error.

Configuración de un documento de índice

Para configurar un documento de índice mediante la consola de S3, utilice el siguiente procedimiento. También puede configurar un documento de índice mediante la API de REST, los SDK de AWS, la AWS CLI o AWS CloudFormation.

> **Note**
>
> En un bucket habilitado para control de versiones, puede cargar varias copias de `index.html`, pero solo se resolverá la versión más reciente. Para obtener más información sobre el uso del control de versiones de S3, consulte Usar el control de versiones en buckets de S3.

Cuando habilite el alojamiento de sitio web estático para su bucket, escriba el nombre del documento de índice (por ejemplo: **`index.html`**). Después de habilitar el alojamiento de sitio web estático para el bucket, cargue un archivo HTML con el nombre de este documento de índice en el bucket.

Para configurar el documento de índice

1. Cree un archivo `index.html`.

   Si no tiene un archivo `index.html`, puede usar el siguiente HTML para crear uno:

   ```
   <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. Guarde el archivo de índice localmente.

   El nombre del archivo de documento de índice debe coincidir exactamente con el nombre del documento de índice que especifique en el cuadro de diálogo Static website hosting (Alojamiento de sitio web estático). El nombre del documento de índice distingue entre mayúsculas y minúsculas. Por ejemplo, si escribe `index.html` en el nombre del Index document (Documento de índice) en el cuadro de diálogo Static website hosting (Alojamiento de sitio web estático), el nombre del archivo de documento de índice también debe ser `index.html` y no `Index.html`.

3. Inicie sesión en la AWS Management Console y abra la consola de Amazon S3 en https://console.aws.amazon.com/s3/.

4. En la lista Buckets, elija el nombre del bucket que desea utilizar para alojar un sitio web estático.

5. Habilite el alojamiento de sitios web estáticos para su bucket e introduzca el nombre exacto del documento de índice (por ejemplo: `index.html`). Para obtener más información, consulte Habilitar el alojamiento de sitios web.

   Después de habilitar el alojamiento estático del sitio web, continúe con el paso 6.
6. Para cargar el documento de índice en el bucket, realice una de las siguientes acciones:

   - Arrastre y suelte el archivo de índice en la lista de buckets de la consola.
   - Elija Upload (Cargar) y siga las instrucciones para elegir y cargar el archivo de índice.

   Para obtener instrucciones paso a paso, consulte Carga de objetos.
7. (Opcional) Cargue otros contenidos del sitio web en su bucket.

A continuación, debe establecer permisos para el acceso al sitio web. Para obtener información, consulte Configurar permisos para el acceso a sitios web.

También puede configurar opcionalmente un documento de error, un registro de tráfico web o una redirección.

Configurar un documento de error personalizado

Después de configurar el bucket como un sitio web estático, cuando se produce un error, Amazon S3 devuelve un documento de error HTML. Opcionalmente, puede configurar el bucket con un documento de error personalizado para que Amazon S3 devuelva dicho documento cuando se produzca un error.

> **Note**
>
> Algunos navegadores muestran su propio mensaje de error cuando se produce un error, y omiten el documento de error de Amazon S3. Por ejemplo, cuando se produce un error HTTP 404 Not Found (HTTP 404 No encontrado), Google Chrome puede omitir el documento de error de Amazon S3 y mostrar su propio error.

Temas

- Códigos de respuesta HTTP de Amazon S3
- Configurar un documento de error personalizado

Códigos de respuesta HTTP de Amazon S3

En la siguiente tabla se muestra el subconjunto de los códigos de respuesta HTTP que Amazon S3 devuelve cuando ocurre un error.

Código de error HTTP	Descripción
301 Moved Permanently (Desplazado permanentemente)	Cuando un usuario envía una solicitud directamente a los puntos de enlace del sitio web de Amazon S3 (`http://s3-website.` *`Region`*`.amazonaws.com/`), Amazon S3 devuelve una respuesta 301 Moved Permanently (301 Trasladado de forma permanente) y redirecciona esas solicitudes a `https://aws.amazon.com/s3/` .
302 Found (Encontrado)	Cuando Amazon S3 recibe una solicitud para una clave `x`, `http://`*`bucket-name`*`.s3-website.` *`Region`*`.amazonaws.com/x` , sin la barra diagonal, primero busca el objeto con el nombre de clave `x`. Si no encuentra el objeto, Amazon S3 determina que la solicitud es para una subcarpeta `x`, redirecciona la solicitud, añade una barra al final y devuelve el código 302 Found (302 Encontrado).
304 Not Modified (No modificado)	Los usuarios de Amazon S3 solicitan encabezados `If-Modified-Since`, `If-Unmodified-Since` , `If-Match` o `If-None-Match` para determinar si el objeto solicitado es el mismo que la copia almacenada que conserva el cliente. Si el objeto es el mismo, el punto de enlace del sitio web devuelve una respuesta 304 Not Modified (304 No modificado).
400 Malformed Request (Solicitud con formato incorrecto)	El punto de enlace de sitio web devuelve una respuesta 400 Malformed Request (Solicitud con formato incorrecto) cuando un usuario intenta obtener acceso a un bucket con el punto de enlace regional incorrecto.
403 Forbidden	El punto de enlace de sitio web devuelve una respuesta 403 Forbidden (403 Prohibido) cuando la solicitud de un usuario se traduce en un objeto que no se puede leer públicamente. El propietario del objeto debe permitir la lectura pública del objeto mediante una política de bucket o una ACL.

Código de error HTTP	Descripción
404 Not Found (No encontrado)	El punto de enlace de sitio web devuelve una respuesta 404 Not Found (404 No encontrado) por las razones siguientes: • Amazon S3 determina que el URL del sitio web hace referencia a una clave de objeto que no existe. • Amazon S3 infiere que la solicitud es para un documento de índice que no existe. • Un bucket especificado en el URL no existe. • Un bucket especificado en el URL existe pero no está configurado como sitio web. Puede crear un documento personalizado que se devuelve para 404 Not Found (404 No encontrado). Asegúrese de que el documento esté cargado al bucket configurado como sitio web y que la configuración de alojamiento del sitio web utilice el documento. Para obtener información acerca de cómo Amazon S3 interpreta un URL como una solicitud para un objeto o un documento de índice, consulte Configurar un documento de índice.
500 Service Error (Error de servicio)	El punto de enlace del sitio web devuelve una respuesta 500 Service Error (500 Error de servicio) cuando ocurre un error interno del servidor.
503 Service Unavailable	El punto de enlace del sitio web devuelve una respuesta 503 Service Unavailable (503 Servicio no disponible) cuando Amazon S3 determina que debe reducir la velocidad de solicitud.

Para cada uno de estos errores, Amazon S3 devuelve un mensaje HTML predefinido. A continuación, se muestra un ejemplo de un mensaje HTML devuelto para la respuesta 403 Forbidden (403 Prohibido).

Configurar un documento de error personalizado

Cuando configura el bucket como un sitio web estático, puede proporcionar un documento de error personalizado que contenga un mensaje de error sencillo y ayuda adicional. Amazon S3 devuelve el documento de error personalizado para las clase de códigos de error HTTP 4XX únicamente.

Para configurar un documento de error personalizado mediante la consola de S3, siga los pasos que se muestran a continuación. También puede configurar un documento de errores mediante la API de REST, los SDK de AWS, la AWS CLI o AWS CloudFormation. Para obtener más información, consulte los siguientes temas:

- PutBucketWebsite en la Referencia de la API de Amazon Simple Storage Service
- AWS::S3::Bucket WebsiteConfiguration en la guía del usuario de AWS CloudFormation
- put-bucket-website en la Referencia de comandos de la AWS CLI

Cuando habilite el alojamiento de sitios webs estáticos para el bucket, escriba el nombre del documento de error (por ejemplo: **`404.html`**). Después de habilitar el alojamiento de sitios web estáticos para el bucket, cargue un archivo HTML con el nombre de este documento de error en el bucket.

Para configurar un documento de error,

1. Cree un documento de error, por ejemplo `404.html`.
2. Guarde el archivo de documento de error localmente.

   El nombre del documento de error distingue mayúsculas y minúsculas y debe coincidir exactamente con el nombre que escriba al habilitar el alojamiento de sitios web estáticos. Por ejemplo, si escribe `404.html` en el nombre del Error document (Documento de error) en el cuadro de diálogo Static website hosting (Alojamiento de sitio web estático), el nombre del archivo del documento de error también debe ser `404.html`.
3. Inicie sesión en la AWS Management Console y abra la consola de Amazon S3 en https://console.aws.amazon.com/s3/.
4. En la lista Buckets, elija el nombre del bucket que desea utilizar para alojar un sitio web estático.
5. Habilite el alojamiento de sitios web estáticos para su bucket y escriba el nombre exacto del documento de error (por ejemplo: `404.html`). Para obtener más información, consulte Habilitar el alojamiento de sitios web y Configurar un documento de error personalizado.

   Después de habilitar el alojamiento estático del sitio web, continúe con el paso 6.
6. Para cargar el documento de error en el bucket, realice una de las siguientes acciones:

   - Arrastre y suelte el archivo del documento de error a la lista de buckets de la consola.
   - Elija Upload (Cargar) y siga las instrucciones para elegir y cargar el archivo de índice.

   Para obtener instrucciones paso a paso, consulte Carga de objetos.

Configurar permisos para el acceso a sitios web

Cuando configura un bucket como un sitio web estático, si desea que el sitio web sea público, puede conceder acceso público de lectura. Para hacer que el bucket sea legible públicamente, debe deshabilitar la configuración de bloqueo de acceso público del bucket y escribir una política de bucket que conceda acceso público de lectura. Si el bucket contiene objetos que no son propiedad del propietario del bucket, es posible que necesite además añadir una lista de control de acceso (ACL) de objeto que conceda acceso de lectura a todo el mundo.

Si no desea desactivar la configuración de acceso público de bloqueo para el bucket pero quiere que su sitio web sea público, puede crear una distribución de Amazon CloudFront para atender su

sitio web estático. Para obtener más información, consulte Aceleración de su sitio web con Amazon CloudFront o Utilizar una distribución de Amazon CloudFront para servir a un sitio web estático en la Guía para desarrolladores de Amazon Route 53.

> **Note**
>
> En el punto de enlace del sitio web, si un usuario solicita un objeto que no existe, Amazon S3 devuelve el código de respuesta HTTP `404 (Not Found)`. Si el objeto existe, pero no se ha concedido el permiso de lectura para él, el punto de enlace del sitio web devuelve el código de respuesta HTTP `403 (Access Denied)`. El usuario puede utilizar el código de respuesta para inferir si existe un objeto específico. Si no desea que esto suceda, no debe activar el soporte de sitio web para el bucket.

Temas

Paso 1: Editar la configuración del S3 Block Public Access

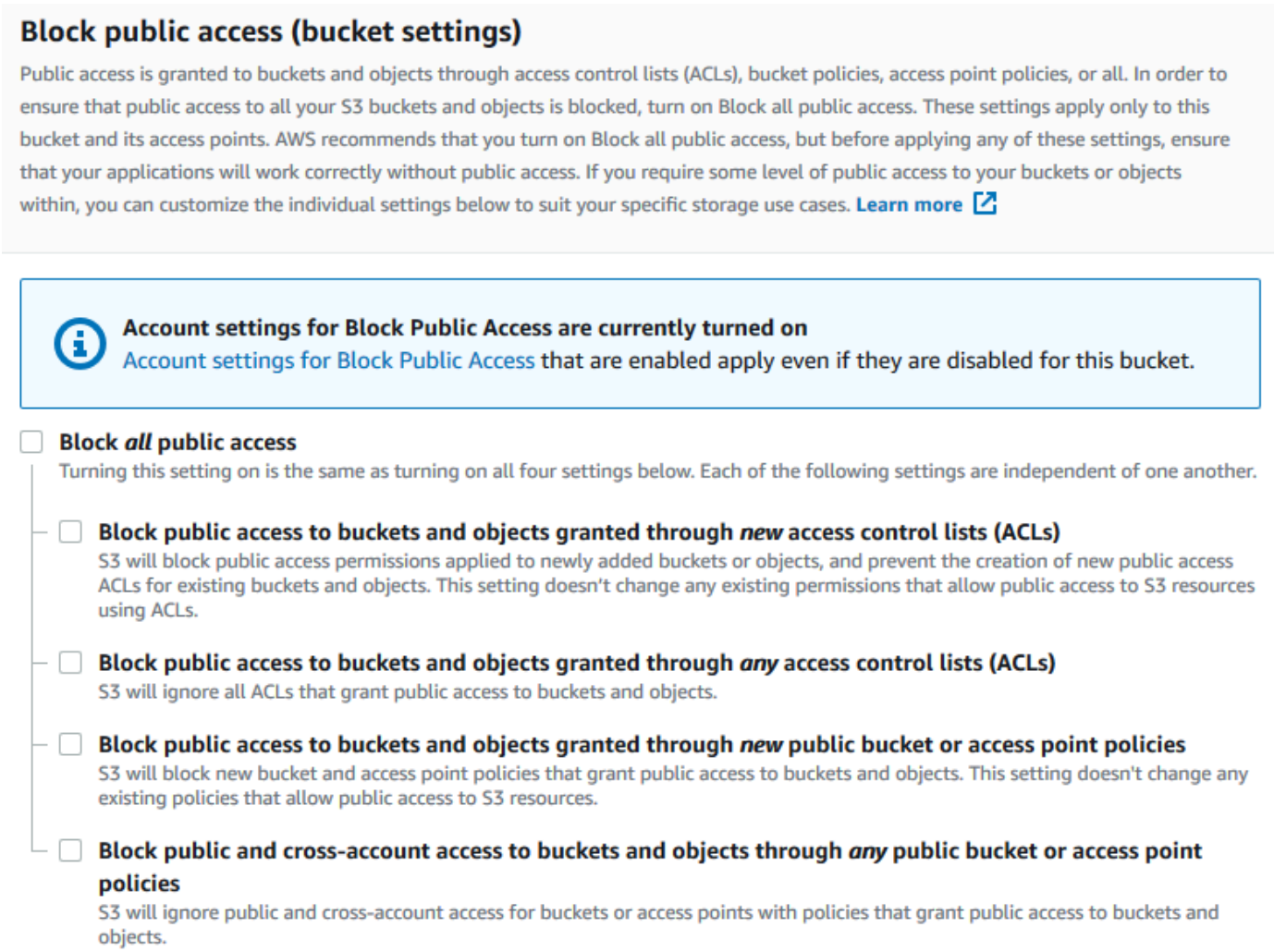
Si desea configurar un bucket existente como sitio web estático que tenga acceso público, debe editar la configuración del bloqueo de acceso público para dicho bucket. También es posible que tenga que editar la configuración del bloqueo de acceso público a nivel de cuenta. Amazon S3 aplica la combinación más restrictiva de la configuración del bloqueo de acceso público a nivel de bucket y nivel de cuenta.

Por ejemplo, si permite el acceso público a un bucket, pero bloquea todo el acceso público en el nivel de cuenta, Amazon S3 seguirá bloqueando el acceso público al bucket. En esta situación, tendría que editar su configuración del bloqueo de acceso público del nivel de bucket y nivel de cuenta. Para obtener más información, consulte Bloquear el acceso público a su almacenamiento de Amazon S3.

De forma predeterminada, Amazon S3 bloquea el acceso público a su cuenta y sus buckets. Si desea utilizar un bucket para alojar un sitio web estático, puede utilizar estos pasos para editar la configuración de bloqueo de acceso público.

> **Warning**
>
> Antes de completar estos pasos, consulte Bloquear el acceso público a su almacenamiento de Amazon S3 para asegurarse de que comprende y acepta los riesgos que implica permitir el acceso público. Cuando desactiva la configuración de acceso público de bloqueo para que el bucket sea público, cualquier usuario de Internet puede acceder al bucket. Le recomendamos que bloquee todo el acceso público a sus buckets.

1. Abra la consola de Amazon S3 en https://console.aws.amazon.com/s3.
2. Elija el nombre del bucket que ha configurado como sitio web estático.
3. Elija Permissions (Permisos).
4. En Block public access (bucket settings) (Bloquear acceso público [configuración de bucket]), elija Edit (Editar).
5. Desactive Block all public access (Bloquear todo el acceso público) y elija Save changes (Guardar cambios).



Amazon S3 desactiva la configuración de Bloqueo de acceso público para su bucket. Para crear un sitio web público y estático, es posible que también tenga que editar la configuración de Bloqueo de acceso público para su cuenta antes de agregar una política de bucket. Si la configuración de Bloqueo de acceso público de su cuenta está activada actualmente, verá una nota en Bloquear acceso público (configuración del bucket).

Paso 2: Agregar una política de bucket

Para hacer que los objetos del bucket sean legibles públicamente, debe escribir una política de bucket que conceda permiso `s3:GetObject` a todo el mundo.

Después de editar la configuración de S3 Block Public Access, debe agregar una política de bucket para garantizar el acceso de lectura público a su bucket. Cuando concede permiso de lectura público, cualquier persona de Internet puede acceder a su bucket.

> **Important**
>
> La política que se muestra a continuación es solo un ejemplo y permite acceso completo al contenido del bucket. Antes de continuar con este paso, revise ¿Cómo puedo proteger los archivos en mi bucket de Amazon S3? para asegurarse de que comprende las prácticas recomendadas para proteger los archivos en el bucket de S3 y los riesgos que implica la concesión de acceso público.

1. En Buckets, elija el nombre del bucket.
2. Elija Permissions (Permisos).
3. En Bucket Policy (Política de bucket), elija Edit (Editar).
4. Para conceder acceso público de lectura a su sitio web, copie la siguiente política de bucket y péguela en el Bucket policy editor (Editor de políticas de bucket).

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Sid": "PublicReadGetObject",
            "Effect": "Allow",
            "Principal": "*",
            "Action": [
                "s3:GetObject"
            ],
            "Resource": [
                "arn:aws:s3:::Bucket-Name/*"
            ]
        }
    ]
}
```

5. Actualice el valor de `Resource` para el nombre de su bucket.

   En la política de bucket de ejemplo anterior, *`Bucket-Name`* es un marcador de posición para el nombre del bucket. Para utilizar esta política de bucket con su propio bucket, debe actualizar este nombre para que coincida con su nombre de bucket.

6. Elija Guardar cambios.

Aparecerá un mensaje que indicará que la política de bucket se ha agregado correctamente.

Si ve un error que indica `Policy has invalid resource`, confirme que el nombre del bucket en la política del bucket coincide con el nombre de su bucket. Para obtener información acerca de cómo agregar una política de bucket, consulte ¿Cómo añado una política de bucket de S3?

Si recibe un mensaje de error y no puede guardar la política de bucket, compruebe la configuración del bloqueo de acceso público para la cuenta y el bucket para confirmar que permite acceso público al bucket.

Listas de control de acceso de objetos

Puede utilizar una política de bucket para conceder permiso de lectura público a los objetos. No obstante, la política de bucket se aplica solo a objetos que pertenecen al propietario del bucket. Si el bucket contiene objetos que no pertenecen al propietario del bucket, este debería utilizar la lista de control de acceso (ACL) del objeto para conceder permiso de LECTURA público en dichos objetos.

S3 Object Ownership es una configuración de bucket de Amazon S3 que puede usar para controlar la propiedad de los objetos que se cargan en el bucket y para activar o desactivar las ACL. De forma predeterminada, la propiedad de objetos se establece en la configuración impuesta por el propietario del bucket. Además, todas las ACL están deshabilitadas. Cuando las ACL están deshabilitadas, el propietario del bucket posee todos los objetos del bucket y administra su acceso de forma exclusiva mediante políticas de administración de acceso.

La mayoría de los casos de uso modernos de Amazon S3 ya no requieren el uso de ACL. Le recomendamos desactivar las ACL, excepto en circunstancias inusuales en las que necesite controlar el acceso a cada objeto de manera individual. Si las ACL están desactivadas, puede usar políticas para controlar el acceso a todos los objetos del bucket, independientemente de quién haya subido los objetos al bucket. Para obtener más información, consulte Control de la propiedad de los objetos y desactivación de las ACL del bucket.

> **Important**
>
> Si el bucket utiliza la configuración de propietario del bucket obligatorio de S3 Object Ownership, debe utilizar políticas para conceder acceso al bucket y a los objetos que contiene. Si la configuración impuesta por el propietario del bucket está activada, las solicitudes de configuración o actualización de las listas de control de acceso (ACL) fallan

y devuelven el código de error `AccessControlListNotSupported`. Las solicitudes de lectura de ACL siguen siendo compatibles.

Para que un objeto sea legible públicamente mediante una ACL, conceda permiso de LECTURA al grupo `AllUsers`, tal como se muestra en el siguiente elemento concedido. Añada el siguiente elemento concedido a la ACL de objetos. Para obtener más información sobre la administración de las ACL, consulte Información general de las Listas de control de acceso (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>
```

(Opcional) Registro del tráfico web

También puede habilitar el registro de acceso al servidor de Amazon S3 para un bucket configurado como un sitio web estático. El registro de acceso al servidor proporciona registros detallados para las solicitudes realizadas a su bucket. Para obtener más información, consulte Registro de solicitudes con registro de acceso al servidor. Si tiene previsto utilizar Amazon CloudFront para acelerar su sitio web, también puede usar el registro de CloudFront. Para obtener más información, consulte Configuración y uso de registros de acceso en la guía para desarrolladores de Amazon CloudFront.

Para habilitar el registro de acceso al servidor para su bucket de sitio web estático

1. Abra la consola de Amazon S3 en https://console.aws.amazon.com/s3.
2. En la misma región en la que creó el bucket configurado como sitio web estático, cree un bucket para el registro, por ejemplo `logs.example.com`.
3. Cree una carpeta para los archivos de registro de acceso al servidor (por ejemplo: `logs`).
4. (Opcional) Si desea utilizar CloudFront para mejorar el rendimiento del sitio web, cree una carpeta para los archivos de registro de CloudFront (por ejemplo: `cdn`).

   Para obtener más información, consulte Aceleración de su sitio web con Amazon CloudFront.
5. En la lista de Buckets, elija el bucket.

6. Seleccione Properties (Propiedades).
7. En Server access logging (Registro de acceso al servidor), elija Edit (Editar).
8. Elija Enable.
9. En el bucket de destino, elija el destino del bucket y la carpeta para los registros de acceso al servidor:

   - Busque la carpeta y la ubicación del bucket:
     1. Elija Browse S3 (Examinar S3).
     2. Elija el nombre del bucket y, a continuación, elija la carpeta de registros.
     3. Elija Choose path (Elegir ruta).
   - Introduzca la ruta del bucket de S3, por ejemplo, **`s3://logs.example.com/logs/`**.
10. Elija Save changes (Guardar cambios).

    En su bucket de registro, ahora puede acceder a sus registros. Amazon S3 escribe los registros de acceso al sitio web en su bucket de registro cada dos horas.

(Opcional) Configuración del redireccionamiento de páginas web

Si su bucket de Amazon S3 está configurado para el alojamiento de sitios web estático, puede configurar el redireccionamiento para el bucket o los objetos que contiene. Tiene las siguientes opciones para configurar el redireccionamiento.

Temas

- Redirigir solicitudes de un punto de enlace de sitio web de su bucket a otro bucket o dominio
- Configurar reglas de redireccionamiento para utilizar redireccionamiento condicional avanzado
- Redirigir solicitudes de un objeto

Redirigir solicitudes de un punto de enlace de sitio web de su bucket a otro bucket o dominio

Puede redirigir todas las solicitudes de un punto de enlace de sitio web de un bucket a otro bucket o dominio. Si redirige todas las solicitudes, las que se hayan realizado al punto de enlace del sitio web se redirigirán al bucket o dominio especificado.

Por ejemplo, si su dominio raíz es `example.com`, y desea enviar solicitudes para `http://example.com` y para `http://www.example.com`, debe crear dos buckets denominados `example.com` y `www.example.com`. A continuación, mantenga el contenido del bucket `example.com` y configure el otro bucket `www.example.com` para redirigir todas las solicitudes al bucket `example.com`. Para obtener más información, consulte Configuración de un sitio web estático mediante un nombre de dominio personalizado.

Para redirigir solicitudes para un punto de enlace de sitio web del bucket

1. Abra la consola de Amazon S3 en https://console.aws.amazon.com/s3.
2. En Buckets (Buckets), elija el nombre del bucket del que desea que procedan las solicitudes redirigidas (por ejemplo, `www.example.com`).
3. Seleccione Properties (Propiedades).
4. Elija Static website hosting (Alojamiento de sitios web estáticos), elija Edit (Editar).
5. Elija Redirect requests for an object (Redirigir solicitudes de un objeto).
6. En el cuadro Host name (Nombre de host) , escriba el punto de enlace de sitio web para el bucket o el dominio personalizado.

   Por ejemplo, si redirigiera las solicitudes a una dirección de dominio raíz, escribiría **`example.com`**.
7. En Protocol (Protocolo), elija el protocolo para las solicitudes redirigidas (none (ninguno),http o https).

   Si no especifica un protocolo, la opción predeterminada es none (ninguno).
8. Elija Save changes (Guardar cambios).

Configurar reglas de redireccionamiento para utilizar redireccionamiento condicional avanzado

Utilizando reglas avanzadas de redireccionamiento, puede dirigir condicionalmente las solicitudes según nombres de clave de objeto, prefijos en la solicitud o códigos de respuesta específicos. Por ejemplo, supongamos que elimina o cambia el nombre de un objeto en el bucket. Puede añadir una regla de enrutamiento que redireccione la solicitud a otro objeto. Si desea que una carpeta no esté disponible, puede añadir una regla de enrutamiento para redirigir la solicitud a otra página web. Además, puede añadir una regla de enrutamiento para gestionar condiciones de error dirigiendo las solicitudes que devuelven un error a otro dominio cuando se procesa el error.

Cuando habilite el alojamiento de sitios web estáticos para el bucket, puede especificar reglas de redireccionamiento avanzadas. Amazon S3 tiene una limitación de 50 reglas de enrutamiento por configuración de sitio web. Si necesita más de 50 reglas de enrutamiento, puede utilizar la redirección de objetos. Para obtener más información, consulte Uso de la consola de S3.

Para obtener más información acerca de la configuración de reglas de enrutamiento mediante la API de REST, consulte PutBucketWebsite en la referencia de la API de Amazon Simple Storage Service.

> **Important**
>
> Para crear reglas de redireccionamiento en la nueva consola de Amazon S3, debe utilizar JSON. Para ver ejemplos de JSON, consulte Ejemplos de reglas de redireccionamiento.

Para configurar reglas de redirección para un sitio web estático

Para agregar reglas de redirección para un bucket que ya tiene habilitado el alojamiento de sitios web estático, siga estos pasos.

1. Abra la consola de Amazon S3 en https://console.aws.amazon.com/s3.
2. En la lista Buckets, elija el nombre del bucket que ha configurado como sitio web estático.
3. Seleccione Properties (Propiedades).
4. Elija Static website hosting (Alojamiento de sitios web estáticos), elija Edit (Editar).
5. En la casilla Redirection rules (Reglas de redireccionamiento), introduzca las reglas de redireccionamiento.

   En la consola de S3, describa las reglas mediante JSON. Para ver ejemplos de JSON, consulte Ejemplos de reglas de redireccionamiento. Amazon S3 tiene una limitación de 50 reglas de enrutamiento por configuración de sitio web.
6. Elija Save changes (Guardar cambios).

Elementos de la regla enrutamiento

Lo que sigue es la sintaxis general para definir las reglas de redireccionamiento en una configuración de sitio web en JSON y XML. Para configurar reglas de redireccionamiento en la nueva consola de S3, debe utilizar JSON. Para ver ejemplos de JSON, consulte Ejemplos de reglas de redireccionamiento.

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.
```

En la siguiente tabla se describen los elementos en la regla de enrutamiento.

Nombre	Descripción
`RoutingRules`	Contenedor de una colección de elementos `RoutingRule` .
`RoutingRule`	Una regla que identifica una condición y el redireccionamiento que se aplica cuando se cumple la condición. Condición: • un contenedor `RoutingRules` debe tener una regla de enrutamiento como mínimo.
`Condition`	Contenedor para describir una condición que debe cumplirse para la aplicación de un redireccionamiento específico. Si la regla de enrutamiento no incluye una condición, esta regla se aplicará a todas las solicitudes.
`KeyPrefixEquals`	El prefijo de un nombre de clave de objeto desde el que se redireccionan las solicitudes. El `KeyPrefixEquals` será necesario si no se especifica `HttpErrorCodeReturnedEquals` . Si se especifican los

Nombre	Descripción
	elementos `KeyPrefixEquals` y `HttpErrorCodeReturnedEquals`, ambos deben ser verdaderos para que se cumpla la condición.
`HttpErrorCodeReturnedEquals`	El código de error HTTP que debe coincidir para que se aplique el redireccionamiento. Si se produce un error y el código del error coincide con este valor, se aplicará el redireccionamiento especificado. El `HttpErrorCodeReturnedEquals` será necesario si no se especifica `KeyPrefixEquals`. Si se especifican los elementos `KeyPrefixEquals` y `HttpErrorCodeReturnedEquals`, ambos deben ser verdaderos para que se cumpla la condición.
`Redirect`	Un elemento del contenedor que provee instrucciones para redireccionar la solicitud. Puede redireccionar solicitudes a otro host o a otra página, o puede especificar el uso de otro protocolo. Cada `RoutingRule` debe tener un elemento `Redirect`. El elemento `Redirect` debe tener uno de los siguientes elementos del mismo nivel como mínimo: `Protocol`, `HostName`, `ReplaceKeyPrefixWith`, `ReplaceKeyWith` o `HttpRedirectCode`.
`Protocol`	El protocolo, `http` o `https`, utilizado en el encabezado `Location` que se devuelve en la respuesta. No se requiere `Protocol`, si se provee uno de sus elementos del mismo nivel.

Nombre	Descripción
`HostName`	El nombre de host utilizado en el encabezado `Location` que se devuelve en la respuesta. No se requiere `HostName`, si se provee uno de sus elementos del mismo nivel.
`ReplaceKeyPrefixWith`	El prefijo del nombre de clave de objeto que sustituye al valor de `KeyPrefixEquals` en la solicitud de redireccionamiento. No se requiere `ReplaceKeyPrefixWith`, si se provee uno de sus elementos del mismo nivel. Solo se puede proveer si no se proporciona `ReplaceKeyWith`.
`ReplaceKeyWith`	La clave del objeto que utilizar en el encabezado `Location` que se devuelve en la respuesta. No se requiere `ReplaceKeyWith`, si se provee uno de sus elementos del mismo nivel. Solo se puede proveer si no se proporciona `ReplaceKeyPrefixWith`.
`HttpRedirectCode`	El código de redireccionamiento de HTTP utilizado en el encabezado `Location` que se devuelve en la respuesta. No se requiere `HttpRedirectCode`, si se provee uno de sus elementos del mismo nivel.

Ejemplos de reglas de redireccionamiento

En los siguientes ejemplos se explican las tareas de redireccionamiento más comunes:

> **Important**
>
> Para crear reglas de redireccionamiento en la nueva consola de Amazon S3, debe utilizar JSON.

Example 1: redireccionamiento luego de cambiar el nombre de un prefijo de clave.

Supongamos que el bucket contiene los siguientes objetos:

- index.html
- docs/article1.html
- docs/article2.html

Ha decidido cambiar el nombre de la carpeta `docs/` a `documents/`. Después de hacer este cambio, deberá redireccionar las solicitudes para el prefijo `docs/` a `documents/`. Por ejemplo, la solicitud para `docs/article1.html` se redireccionará a `documents/article1.html`.

En este caso, debe añadir la siguiente regla de enrutamiento a la configuración del sitio 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: redireccionamiento de solicitudes de una carpeta eliminada a una página.

Supongamos que elimina la carpeta `images/` (es decir, que elimina todos los objetos con el prefijo de clave `images/`). Puede añadir una regla de enrutamiento que redireccione las solicitudes de cualquier objeto con el prefijo de clave `images/` a una página denominada `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: redireccionamiento de error de HTTP.

Supongamos que cuando no se encuentra un objeto solicitado, desea redireccionar las solicitudes a una instancia Amazon Elastic Compute Cloud (Amazon EC2). Debe añadir una regla de redireccionamiento para que, cuando se devuelva el código de estado 404 (Not Found) [404 (No encontrado)], se redireccione al visitante del sitio a una instancia Amazon EC2 que se encarga de la solicitud.

En el siguiente ejemplo, también se inserta el prefijo de clave de objeto `report-404/` en el redireccionamiento. Por ejemplo, si solicita la página `ExamplePage.html` y se traduce en un error HTTP 404, la solicitud se redirecciona a la página `report-404/ExamplePage.html` en la instancia Amazon EC2 especificada. Si no hay una regla de enrutamiento y se produce un error HTTP 404, se devolverá el documento de error especificado en la configuración.

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>
```

Redirigir solicitudes de un objeto

Puede redirigir las solicitudes de un objeto a otro objeto o URL estableciendo la ubicación de redirección del sitio web en los metadatos del objeto. Para configurar el redireccionamiento, debe añadir la propiedad `x-amz-website-redirect-location` a los metadatos del objeto. En la consola de Amazon S3, puede establecer la ubicación del redireccionamiento de un sitio web en los

metadatos del objeto. Si utiliza la API de Amazon S3, se establece `x-amz-website-redirect-location`. Luego, el sitio web interpreta el objeto como un redireccionamiento 301.

Para redireccionar una solicitud a otro objeto, debe establecer la ubicación de redireccionamiento para la clave del objeto de destino. Para redireccionar una solicitud a un URL externo, debe establecer la ubicación de redireccionamiento para el URL que desee. Para obtener más información acerca de los metadatos del objeto, consulte Metadatos de objetos definidos por el sistema.

Cuando establece un redireccionamiento de página, puede conservar o eliminar el contenido del objeto de destino. Por ejemplo, si tiene un objeto `page1.html` en el bucket, puede redirigir cualquier solicitud de esta página a otro objeto, `page2.html`. Dispone de dos opciones para hacerlo:

- Mantenga el contenido del objeto `page1.html` y redirija las solicitudes de página.
- Elimine el contenido de `page1.html` y cargue un objeto de cero bytes denominado `page1.html` para reemplazar el objeto existente y redirigir las solicitudes de página.

Uso de la consola de S3

1. Abra la consola de Amazon S3 en https://console.aws.amazon.com/s3.
2. En la lista Buckets, elija el nombre del bucket que ha configurado como sitio web estático (por ejemplo: `example.com`).
3. En Objects (Objetos), seleccione su objeto.
4. Elija Actions (Acciones), y elija Edit metadata (Editar metadatos).
5. Elija Metadata (Metadatos).
6. Elija Add Metadata (Añadir metadatos).
7. En Typo (Tipo), elija System Defined (Definido por el sistema).
8. En Key (Clave), elija x-amz-website-redirect-location.
9. En Value (Valor), introduzca el nombre de clave del objeto que desee redirigir, por ejemplo, `/page2.html`.

   Para otro objeto del mismo bucket, se requiere el prefijo `/` del valor. También puede establecer el valor para un URL externo, por ejemplo, `http://www.example.com`.
10. Elija Edit medatada (Editar metadatos).

Uso de la API de REST

Las siguientes acciones de la API de Amazon S3 admiten el encabezado `x-amz-website-redirect-location` en la solicitud. Amazon S3 almacena el valor del encabezado en los metadatos del objeto como `x-amz-website-redirect-location`.

- PUT Object
- Initiate Multipart Upload
- POST Object
- PUT Object - Copy

Un bucket configurado para un alojamiento de sitio web contiene el punto de enlace de sitio web y el punto de enlace REST. Una solicitud para una página que está configurada como un redireccionamiento 301 tiene los siguientes resultados posibles, en función del punto de enlace de la solicitud:

- Punto de enlace de sitio web en región específica: Amazon S3 redirecciona la solicitud de la página según el valor de la propiedad `x-amz-website-redirect-location`.
- Punto de enlace REST: Amazon S3; no redirecciona la solicitud de la página. Devuelve el objeto solicitado.

Para obtener más información acerca de los puntos de enlace, consulte Diferencias clave entre el punto de enlace de un sitio web y un punto de enlace de la API de REST.

Cuando establece un redireccionamiento de página, puede conservar o eliminar el contenido del objeto. Por ejemplo, supongamos que tiene un objeto `page1.html` en el bucket.

- Para conservar el contenido de `page1.html` y redirigir solo las solicitudes de página, puede enviar una solicitud PUT Object - Copy para crear un nuevo objeto `page1.html` que utilice el objeto existente `page1.html` como origen. Debe establecer el encabezado `x-amz-website-redirect-location` en su solicitud. Cuando se completa la solicitud, la página original tendrá su contenido sin cambios, pero Amazon S3 redireccionará cualquier solicitud a esa página a la ubicación de redireccionamiento que especificó.
- Para eliminar el contenido del objeto `page1.html` y redireccionar solicitudes a la página, puede enviar una solicitud PUT Object para cargar un objeto de cero bytes con la misma clave de objeto: `page1.html`. En la solicitud PUT, debe establecer `x-amz-website-redirect-location` para `page1.html` para el nuevo objeto. Cuando se completa la solicitud, `page1.html` no

tendrá contenido y las solicitudes serán redireccionadas a la ubicación especificada por `x-amz-website-redirect-location`.

Cuando recupera el objeto con la acción GET Object, junto con los metadatos de otro objeto, Amazon S3 devuelve el encabezado `x-amz-website-redirect-location` en la respuesta.

Uso compartido de recursos entre orígenes (CORS)

El uso compartido de recursos entre orígenes (CORS) define una manera para que las aplicaciones web de los clientes cargadas en un dominio interactúen con los recursos de un dominio diferente. Con el soporte del CORS, puede desarrollar aplicaciones web del lado del cliente completas con Amazon S3 y permitir un acceso entre orígenes a sus recursos de Amazon S3 de manera selectiva.

En esta sección, se proporciona información general acerca del CORS. En los subtemas se describe cómo puede habilitar el CORS con la consola de Amazon S3 o mediante programación utilizando la API de REST de Amazon S3 y los SDK de AWS.

Compartir recursos entre orígenes: escenarios de casos de uso

A continuación, se muestran ejemplos de casos de uso de CORS:

Escenario 1

Suponga que aloja un sitio web en un bucket de Amazon S3 llamado `website` como se describe en Alojamiento de un sitio web estático mediante Amazon S3. Sus usuarios cargan el punto de conexión de sitio web.

```
http://website.s3-website.us-east-1.amazonaws.com
```

Ahora desea usar el código JavaScript en la páginas web que se almacenan en este bucket para poder realizar solicitudes GET y PUT autenticadas con el mismo bucket mediante el punto de conexión de la API de Amazon S3 para el bucket, `website.s3.us-east-1.amazonaws.com`. Un navegador, por lo general, bloquearía el código JavaScript y no permitiría realizar esas solicitudes, pero con el CORS puede configurar el bucket para que se habiliten de manera explícita las solicitudes entre orígenes de `website.s3-website.us-east-1.amazonaws.com`.

Escenario 2

Suponga que desea alojar una fuente web en su bucket de S3. Nuevamente, los navegadores requieren una comprobación del CORS (también conocida como comprobación preliminar) para la carga de fuentes web, por lo que configuraría el bucket que aloja la fuente web para que permita que cualquier origen realice estas solicitudes

¿Cómo evalúa Amazon S3 la configuración de CORS en un bucket?

Cuando Amazon S3 recibe una solicitud de comprobación preliminar de un navegador, evalúa la configuración de CORS para el bucket y usa la primera regla `CORSRule` que coincide con la solicitud del navegador entrante para habilitar una solicitud entre orígenes. Para que una regla coincida, se deben cumplir las siguientes condiciones:

- El encabezado `Origin` de una solicitud de CORS a su bucket debe corresponderse con los orígenes del elemento `AllowedOrigins` de su configuración de CORS.
- Los métodos HTTP que se especifiquen en el `Access-Control-Request-Method` en una solicitud de CORS a su bucket deben corresponderse con el método o métodos enumerados en el elemento `AllowedMethods` de su configuración de CORS.
- Los encabezados que aparecen en el encabezado `Access-Control-Request-Headers` de una solicitud anterior al tránsito deben corresponderse con los encabezados del elemento `AllowedHeaders` de la configuración de CORS.

> **Note**
>
> Las ACL y políticas siguen aplicándose cuando habilita CORS en su bucket.

Cómo el punto de acceso de Object Lambda da soporte a CORS

Cuando S3 Object Lambda recibe una solicitud de un navegador o la solicitud incluye un encabezado `Origin`, S3 Object Lambda siempre añade un campo de encabezado `“AllowedOrigins":"*"`.

Para obtener más información acerca del uso de CORS, consulte los temas siguientes.

Temas

- Elementos de una configuración de CORS
- Configuración del uso compartido de recursos entre orígenes (CORS)
- Solución de problemas de CORS

Elementos de una configuración de CORS

Para configurar su bucket para permitir solicitudes entre orígenes, debe crear una configuración CORS. La configuración de CORS es un documento con elementos que identifican los orígenes desde los que se permitirá el acceso al bucket, las operaciones (métodos HTTP) que permitirá para cada origen y otro tipo de información específica de cada operación. Puede añadir hasta 100 reglas a la configuración. Puede agregar la configuración de CORS como recurso secundario `cors` al bucket.

Si está configurando CORS en la consola S3, debe utilizar JSON para crear una configuración CORS. La nueva consola de S3 solo admite configuraciones de CORS JSON.

Para obtener más información acerca de la configuración de CORS y los elementos que contiene, consulte los temas siguientes. Para obtener instrucciones sobre cómo agregar una configuración CORS, consulte Configuración del uso compartido de recursos entre orígenes (CORS).

> **Important**
>
> En la consola de S3, la configuración de CORS debe ser JSON.

Temas

Elemento **`AllowedMethods`**

En la configuración CORS, puede especificar los siguientes valores para el elemento `AllowedMethods`:

- GET
- PUT
- POST

- ELIMINAR
- HEAD

Elemento **AllowedOrigins**

En el elemento `AllowedOrigins`, se especifican los orígenes desde los cuales desea permitir solicitudes entre dominios, por ejemplo, `http://www.example.com`. La cadena de origen solo puede contener un carácter comodín `*`, como `http://*.example.com`. Como alternativa, se puede especificar el carácter `*` como el origen para habilitar el envío de solicitudes entre orígenes de todos los orígenes. Se puede también especificar el método `https` para solo habilitar orígenes seguros.

Elemento **AllowedHeaders**

El elemento `AllowedHeaders` especifica qué encabezados se permiten en una solicitud de comprobación preliminar a través del encabezado `Access-Control-Request-Headers`. Cada nombre de encabezado en el encabezado `Access-Control-Request-Headers` debe coincidir con una entrada correspondiente en el elemento. Amazon S3 solo envía los encabezados permitidos en una respuesta que se solicitaron. Para ver una lista de muestra de los encabezados que se pueden usar en las solicitudes que se envían a Amazon S3, consulte el tema sobre encabezados de solicitud comunes en la guía de referencia de API de Amazon Simple Storage Service.

Cada cadena AllowedHeaders de la configuración puede contener como máximo un carácter comodín *. Por ejemplo, el elemento `<AllowedHeader>x-amz-*</AllowedHeader>` habilita los encabezados exclusivos de Amazon.

Elemento **ExposeHeaders**

Cada elemento `ExposeHeader` identifica un encabezado en la respuesta al que desea que los clientes puedan obtener acceso desde sus aplicaciones (por ejemplo, desde un objeto `XMLHttpRequest` con código JavaScript). Para ver una lista de encabezados de respuesta comunes de Amazon S3, vaya al tema sobre encabezados de respuesta comunes en la guía de referencia de API de Amazon Simple Storage Service .

Elemento **MaxAgeSeconds**

El elemento `MaxAgeSeconds` especifica el tiempo en segundos que el navegador puede almacenar en caché la respuesta de una solicitud de comprobación preliminar según la identifica el recurso, el método HTTP y el origen.

Ejemplos de configuraciones de CORS

En lugar de obtener acceso a un sitio web con el punto de conexión del sitio web de Amazon S3, puede usar su propio dominio, como `example1.com`, para servir su contenido. Para obtener información acerca de su propio dominio, consulte Tutorial: Configuración de un sitio web estático mediante un dominio personalizado registrado con Route 53.

En el siguiente ejemplo, la configuración de CORS tiene tres reglas, que se especifican como elementos `CORSRule`:

- La primera regla permite solicitudes PUT, POST y DELETE entre orígenes del origen `http://www.example1.com`. La regla también permite todos los encabezados en una solicitud OPTIONS de comprobación preliminar a través del encabezado `Access-Control-Request-Headers`. Como respuesta a solicitudes OPTIONS de comprobación preliminar, Amazon S3 devuelve los encabezados solicitados.
- La segunda regla permite las mismas solicitudes entre orígenes que la primera, pero se aplica a otro origen, `http://www.example2.com`.
- La tercera regla permite solicitudes GET entre orígenes de todos los orígenes. El carácter comodín `*` hace referencia a todos los orígenes.

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>
```

La configuración CORS también permite parámetros de configuración opcionales, como se muestra en la siguiente configuración CORS. En este ejemplo, la configuración CORS permite las solicitudes PUT, POST y DELETE entre orígenes desde el origen `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>
```

El elemento `CORSRule` en la configuración anterior incluye los siguientes elementos opcionales:

- `MaxAgeSeconds`: especifica la cantidad de tiempo en segundos (en este ejemplo, 3000) que el navegador almacena en caché una respuesta de Amazon S3 a una solicitud OPTIONS de comprobación preliminar para el recurso especificado. Mediante el almacenamiento en caché de la respuesta, el navegador no tiene que enviar solicitudes de comprobación preliminar a Amazon S3 si se repetirá la solicitud original.
- `ExposeHeader`: identifica los encabezados de respuesta (en este ejemplo, `x-amz-server-side-encryption`, `x-amz-request-id` y `x-amz-id-2`) a los que los clientes pueden obtener acceso desde sus aplicaciones (por ejemplo, desde un objeto `XMLHttpRequest` con código JavaScript).

Configuración del uso compartido de recursos entre orígenes (CORS)

El uso compartido de recursos entre orígenes (CORS) define una manera para que las aplicaciones web de los clientes cargadas en un dominio interactúen con los recursos de un dominio diferente. Con el soporte del CORS, puede desarrollar aplicaciones web del lado del cliente completas con Amazon S3 y permitir un acceso entre orígenes a sus recursos de Amazon S3 de manera selectiva.

En esta sección, se muestra cómo habilitar el CORS con la consola de Amazon S3, la API de REST de Amazon S3 y los SDK de AWS. Para configurar el bucket con la finalidad de permitir solicitudes entre orígenes, agregue una configuración CORS al bucket. Una configuración CORS es un documento que define reglas que identifican los orígenes desde los que permitirá el acceso a su bucket, las operaciones (métodos HTTP) permitidas para cada origen y otro tipo de información específica a cada operación. En la consola de S3, la configuración de CORS debe ser un documento JSON.

Para obtener ejemplos de configuraciones CORS en JSON y XML, consulte Elementos de una configuración de CORS.

Uso de la consola de S3

En esta sección también se explica cómo utilizar la consola de Amazon S3 para agregar una configuración de uso compartido de recursos entre orígenes (CORS) a un bucket de S3.

Cuando activa CORS en el bucket, la lista de control de acceso (ACL) y otras políticas de permisos para la obtención de accesos seguirán aplicándose.

> **Important**
>
> En la consola de S3, la configuración de CORS debe ser JSON. Para obtener ejemplos de configuraciones CORS en JSON y XML, consulte Elementos de una configuración de CORS.

Para agregar una configuración CORS a un bucket de S3

1. Inicie sesión en la AWS Management Console y abra la consola de Amazon S3 en https://console.aws.amazon.com/s3/.
2. En la lista Buckets (Buckets), seleccione el nombre del bucket para el que desea crear una política de bucket.
3. Elija Permissions.
4. En la sección Cross-origin resource sharing (CORS) [Compartir recursos entre orígenes (CORS)], elija Edit (Editar).
5. En la casilla de texto CORS configuration editor (Editor de configuración de CORS), escriba o copie y pegue una nueva configuración CORS o edite una que ya exista.

   La configuración CORS es un archivo JSON. El texto que escriba en el editor debe tener un formato JSON válido. Para obtener más información, consulte Elementos de una configuración de CORS.
6. Elija Save changes.

   > **Note**
   >
   > Amazon S3 muestra el nombre de recurso de Amazon (ARN) del bucket junto al título CORS configuration editor (Editor de configuración de CORS). Para obtener más información sobre los nombres ARN, consulte Nombres de recursos de Amazon (ARN)

y espacios de nombres de servicios de AWS en la Referencia general de Amazon Web Services.

Uso de los AWS SDK

Puede usar el AWS SDK para administrar el uso compartido de recursos entre orígenes (CORS) de un bucket. Para obtener más información acerca de CORS, consulte Uso compartido de recursos entre orígenes (CORS).

Los siguientes ejemplos:

- Crea una configuración CORS y establece la configuración en un bucket
- Recupera la configuración y la modifica añadiendo una regla
- Añade la configuración modificada al bucket
- Elimina configuración

Java

Example

Example

Para obtener instrucciones sobre cómo crear y probar una muestra funcional, consulte Introducción en la Guía del desarrollador de AWS SDK for Java.

```
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

Para obtener información acerca de cómo configurar y ejecutar ejemplos de código, consulte Introducción al SDK de AWS para .NET en la Guía para desarrolladores del SDK de AWS para .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()));
            }
        }
    }
}
```

Uso de la API de REST

Para establecer una configuración de CORS en su bucket, puede usar la AWS Management Console. Si la aplicación lo requiere, puede también enviar solicitudes REST directamente. En las secciones siguientes de la referencia de la API de Amazon Simple Storage Service se describen las acciones de la API de REST relacionadas con la configuración de CORS.

- PutBucketCors
- GetBucketCors
- DeleteBucketCors
- OPTIONS object

Solución de problemas de CORS

Los siguientes temas pueden ayudarle a solucionar problemas habituales de CORS relacionados con S3.

Temas

- Error 403 Forbidden: CORS is not enabled for this bucket
- Error 403 Forbidden: This CORS request is not allowed
- No se encontraron encabezados en la respuesta de CORS
- Consideraciones sobre CORS en las integraciones del proxy de S3

Error 403 Forbidden: CORS is not enabled for this bucket

El siguiente error `403 Forbidden` se produce cuando se envía una solicitud entre orígenes a Amazon S3, pero CORS no está configurado en el bucket de S3.

Error: HTTP/1.1 403 Forbidden CORS Response: CORS is not enabled for this bucket.

La configuración de CORS es un documento o política con reglas que identifican los orígenes desde los que se permitirá el acceso al bucket, las operaciones (métodos HTTP) que permitirá para cada origen y otro tipo de información específica de cada operación. Descubra cómo configurar CORS en S3 utilizando la consola de Amazon S3, los AWS SDK y la API de REST. Para obtener más información sobre CORS y ejemplos de una configuración de CORS, consulte Elementos de CORS.

Error 403 Forbidden: This CORS request is not allowed

Se recibe el siguiente error `403 Forbidden` cuando una regla de CORS de la configuración de CORS no se corresponde con los datos de la solicitud.

Error: HTTP/1.1 403 Forbidden CORS Response: This CORS request is not allowed.

Como resultado, este error `403 Forbidden` puede producirse por varios motivos:

- El origen no está permitido.
- Los métodos no están permitidos.
- Los encabezados solicitados no están permitidos.

Para cada solicitud que Amazon S3 reciba, debe tener una regla de CORS en la configuración de CORS que se corresponda con los datos de la solicitud.

El origen no está permitido

El encabezado `Origin` de una solicitud de CORS a su bucket debe corresponderse con los orígenes del elemento `AllowedOrigins` de su configuración de CORS. Un carácter comodín (`"*"`) en el elemento `AllowedOrigins` se correspondería con todos los métodos HTTP. Para obtener más información sobre cómo actualizar el elemento `AllowedOrigins`, consulte Configuración del uso compartido de recursos entre orígenes (CORS).

Por ejemplo, si solo se incluye el dominio `http://www.example1.com` en el elemento `AllowedOrigins`, una solicitud de CORS enviada desde el dominio `http://www.example2.com` recibirá el error `403 Forbidden`.

El siguiente ejemplo muestra parte de una configuración de CORS que incluye el dominio `http://www.example1.com` en el elemento `AllowedOrigins`.

```
"AllowedOrigins":[
   "http://www.example1.com"
]
```

Para que una solicitud de CORS enviada desde el dominio `http://www.example2.com` se realice correctamente, el dominio `http://www.example2.com` debe incluirse en el elemento `AllowedOrigins` de la configuración de CORS.

```
"AllowedOrigins":[
   "http://www.example1.com"
   "http://www.example2.com"
]
```

Los métodos no están permitidos

Los métodos HTTP que se especifiquen en el `Access-Control-Request-Method` en una solicitud de CORS a su bucket deben corresponderse con el método o métodos enumerados en el elemento `AllowedMethods` de su configuración de CORS. Un carácter comodín (`"*"`) en `AllowedMethods` se correspondería con todos los métodos HTTP. Para obtener más información sobre cómo actualizar el elemento `AllowedOrigins`, consulte Configuración del uso compartido de recursos entre orígenes (CORS).

En una configuración de CORS, puede especificar los siguientes métodos en el elemento `AllowedMethods`:

- `GET`
- `PUT`
- `POST`
- `DELETE`
- `HEAD`

El siguiente ejemplo muestra parte de una configuración de CORS que incluye el método `GET` en el elemento `AllowedMethods`. Solo se aceptarán las solicitudes que incluyan el método `GET`.

```
"AllowedMethods":[
```

```
   "GET"
]
```

Si se utilizó un método HTTP (por ejemplo, PUT) en una solicitud de CORS o se incluyó en una solicitud de CORS anterior al tránsito en su bucket, pero el método no está presente en la configuración de CORS, la solicitud dará lugar a un error `403 Forbidden`. Para permitir esta solicitud de CORS o una solicitud de CORS anterior al tránsito, debe agregar el método PUT a su configuración de CORS.

```
"AllowedMethods":[
   "GET"
   "PUT"
]
```

Los encabezados solicitados no están permitidos

Los encabezados que aparecen en el encabezado `Access-Control-Request-Headers` de una solicitud anterior al tránsito deben corresponderse con los encabezados del elemento `AllowedHeaders` de la configuración de CORS. Para obtener una lista de los encabezados comunes que pueden utilizarse en las solicitudes a Amazon S3, consulte Common Request Headers. Para obtener más información sobre cómo actualizar el elemento `AllowedHeaders`, consulte Configuración del uso compartido de recursos entre orígenes (CORS).

El siguiente ejemplo muestra parte de una configuración de CORS que incluye el encabezado `Authorization` en el elemento `AllowedHeaders`. Solo se aceptarían solicitudes del encabezado `Authorization`.

```
"AllowedHeaders":  [
    "Authorization"
]
```

Si se incluyó un encabezado (por ejemplo, `Content-MD5`) en una solicitud de CORS, pero el encabezado no está presente en la configuración de CORS, la solicitud dará lugar a un error `403 Forbidden`. Para permitir esta solicitud de CORS, se debe agregar el encabezado `Content-MD5` a la configuración de CORS. Si quiere pasar los dos encabezados, `Authorization` y `Content-MD5`, de una solicitud de CORS a su bucket, confirme que ambos estén incluidos en el elemento `AllowedHeaders` de su configuración de CORS.

```
"AllowedHeaders":  [
```

```
    "Authorization"
    "Content-MD5"
]
```

No se encontraron encabezados en la respuesta de CORS

El elemento `ExposeHeaders` de su configuración de CORS identifica los encabezados de respuesta que desea que sean accesibles a los scripts y las aplicaciones que se ejecutan en los navegadores en respuesta a una solicitud de CORS.

Si los objetos almacenados en el bucket de S3 tienen metadatos definidos por el usuario (por ejemplo, `x-amz-meta-custom-header`) junto con los datos de respuesta, este encabezado personalizado podría contener metadatos o información adicionales a los que desee acceder desde el código JavaScript del cliente. Sin embargo, de forma predeterminada, los navegadores bloquean el acceso a los encabezados personalizados por motivos de seguridad. Para permitir que el JavaScript del cliente acceda a los encabezados personalizados, debe incluir el encabezado en la configuración de CORS.

En el siguiente ejemplo, el encabezado `x-amz-meta-custom-header1` se incluye en el elemento `ExposeHeaders`. `x-amz-meta-custom-header2` no está incluido en el elemento `ExposeHeaders` y no aparece en la configuración de CORS. En la respuesta, solo se devolverían los valores incluidos en el elemento `ExposeHeaders`. Si la solicitud incluía el encabezado `x-amz-meta-custom-header2` en el encabezado `Access-Control-Expose-Headers`, la respuesta seguiría devolviendo un `200 OK`. Sin embargo, solo devolvería y se mostraría en la respuesta el encabezado permitido (por ejemplo, `x-amz-meta-custom-header`).

```
"ExposeHeaders":  [
    "x-amz-meta-custom-header1"
]
```

Para garantizar que todos los encabezados aparezcan en la respuesta, agregue todos los encabezados permitidos al elemento `ExposeHeaders` en su configuración de CORS, como se muestra a continuación.

```
"ExposeHeaders":  [
    "x-amz-meta-custom-header1",
    "x-amz-meta-custom-header2"
]
```

Consideraciones sobre CORS en las integraciones del proxy de S3

Si se producen errores y ya ha comprobado la configuración de CORS en el bucket de S3 y la solicitud entre orígenes se envía a proxies como AWS CloudFront, intente lo siguiente:

- Configure los ajustes para permitir el método `OPTIONS` para las solicitudes HTTP.
- Configure el proxy para reenviar los siguientes encabezados: `Origin`, `Access-Control-Request-Headers` y `Access-Control-Request-Method`.

Algunos proxies proporcionan características predefinidas para las solicitudes de CORS. Por ejemplo, en CloudFront, puede configurar una política que incluya los encabezados que permiten las solicitudes de CORS cuando el origen es un bucket de S3. Para obtener más información, consulte Control de las solicitudes de origen con una política o Uso de políticas de solicitudes de origen administradas en la Guía para desarrolladores de Amazon CloudFront.

Desarrollo con Amazon S3

En esta sección se tratan temas relacionados con los desarrolladores en torno al uso de Amazon S3. Para obtener más información, consulte los siguientes temas.

> **Note**
>
> Para obtener más información sobre el uso de la clase de almacenamiento Amazon S3 Express One Zone con buckets de directorio, consulte ¿Qué es S3 Express One Zone? y Buckets de directorio.

Temas

Realizar solicitudes

Amazon S3 es un servicio de REST. Puede enviar solicitudes a Amazon S3 con la API REST o las bibliotecas de encapsulamiento de los SDK de AWS (consulte Código de muestra y bibliotecas) que incluyen la API REST de Amazon S3 subyacente, lo que simplifica sus tareas de programación.

Toda interacción con Amazon S3 es o autenticada o anónima. La autenticación es un proceso que permite verificar la identidad del solicitante que intenta acceder a un producto de Amazon Web Services (AWS). Las solicitudes autenticadas deben incluir un valor de firma que autentique al remitente de la solicitud. El valor de firma, en parte, se genera a partir de las claves de acceso de AWS del solicitante (ID de clave de acceso y clave de acceso secreta). Para obtener más información acerca de cómo obtener claves de acceso, consulte ¿Cómo obtengo credenciales de seguridad? en la Referencia general de AWS

Si utiliza el SDK de AWS, las bibliotecas computan la firma a partir de las claves que usted proporciona. Sin embargo, si realiza llamadas directas a la API REST en su aplicación, debe escribir el código para computar la firma y añadirla a la solicitud.

Temas

Acerca de las claves de acceso

En las siguientes secciones se revisan los tipos de claves de acceso que puede utilizar para realizar solicitudes autenticadas.

Claves de acceso de la Cuenta de AWS

Las claves de acceso de la cuenta proporcionan acceso completo a los recursos de AWS que son propiedad de la cuenta. A continuación, se proporcionan ejemplos de claves de acceso:

- ID de clave de acceso (una cadena alfanumérica de 20 caracteres). Por ejemplo: AKIAIOSFODNN7EXAMPLE
- Clave de acceso secreta (una cadena de 40 caracteres). Por ejemplo: wJalrXUtnFEMI/K7MDENG/bPxRfiCYEXAMPLEKEY

El ID de clave de acceso identifica una Cuenta de AWS de forma exclusiva. Puede utilizar estas claves de acceso para enviar solicitudes autenticadas a Amazon S3.

Claves de acceso de usuario de IAM

Puede crear una Cuenta de AWS para su empresa. Sin embargo, puede haber varios empleados en la organización que necesitan acceso a los recursos de AWS de la organización. Compartir sus claves de acceso de la Cuenta de AWS reduce la seguridad y crear Cuentas de AWS individuales para cada empleado puede no ser conveniente. Además, no puede compartir fácilmente recursos como buckets y objetos ya que estos pertenecen a diferentes cuentas. Para compartir recursos, debe conceder permisos, lo que implica un trabajo adicional.

En dichos casos, puede utilizar AWS Identity and Access Management (IAM) para crear usuarios en la Cuenta de AWS con sus propias claves de acceso y adjuntar políticas de usuario de IAM que concedan los permisos de acceso a recursos adecuados a esos usuarios. Para administrar mejor estos usuarios, IAM le permite crear grupos de usuarios y conceder permisos a nivel grupal que se aplican a todos los usuarios de ese grupo.

Estos usuarios se denominan usuarios de IAM, y usted los crea y administra en AWS. La cuenta principal controla la capacidad de un usuario para acceder a AWS. Cualquier recurso creado por un usuario de IAM se encuentra bajo control de la Cuenta de AWS principal, que paga dicho recurso. Estos usuarios de IAM pueden enviar solicitudes autenticadas a Amazon S3 con sus propias credenciales de seguridad. Para obtener más información acerca de la creación y la administración de usuarios en su Cuenta de AWS, consulte la página de detalles del producto de AWS Identity and Access Management.

Credenciales de seguridad temporales

Además de crear usuarios de IAM con sus propias claves de acceso, IAM también le permite otorgar credenciales de seguridad temporales (claves de acceso temporales y un token de seguridad) a cualquier usuario de IAM para que tenga acceso a sus servicios y recursos de AWS. También puede administrar usuarios en su sistema que no pertenecen a AWS. Estos se conocen como usuarios federados. Además, los usuarios pueden ser aplicaciones que usted crea para acceder a sus recursos de AWS.

IAM proporciona la API de AWS Security Token Service para que pueda solicitar credenciales de seguridad temporales. Para solicitar estas credenciales, puede utilizar la API de STS de AWS o el SDK de AWS. La API devuelve credenciales de seguridad temporales (ID de clave de acceso y clave de acceso secreta) y un token de seguridad. Estas credenciales son válidas solo durante el tiempo que usted especifica cuando las solicita. Usted utiliza el ID de clave de acceso y la clave secreta de la misma manera que los utiliza cuando envía solicitudes con sus claves de acceso de usuario de IAM o su Cuenta de AWS. Además, debe incluir el token en cada solicitud que envía a Amazon S3.

Un usuario de IAM puede solicitar estas credenciales de seguridad temporales para uso propio o puede entregarlas a usuarios federados o aplicaciones. Cuando solicita credenciales de seguridad temporales para usuarios federados, debe proporcionar un nombre de usuario y una política de IAM que defina los permisos que desea asociar a estas credenciales de seguridad temporales. El usuario federado no puede tener más permisos que el usuario de la cuenta principal de IAM que solicitó las credenciales temporales.

Puede utilizar estas credenciales de seguridad temporales para realizar solicitudes a Amazon S3. Las bibliotecas de la API computan el valor de firma necesario con esas credenciales para autenticar su solicitud. Si envía las solicitudes con las credenciales vencidas, Amazon S3 deniega la solicitud.

Para obtener información acerca de cómo firmar solicitudes con credenciales de seguridad temporales en sus solicitudes de API REST, consulte Firmar y autenticar las solicitudes REST. Para obtener más información acerca del envío de solicitudes con los SDK de AWS, consulte Realización de solicitudes con los SDK de AWS.

Para obtener más información acerca de la compatibilidad de IAM con las credenciales de seguridad temporales, consulte Credenciales de seguridad temporales en la guía de usuario de IAM.

Para más seguridad, puede solicitar la Multifactor Authentication (MFA, Autenticación multifactor) cuando accede a sus recursos de Amazon S3 mediante la configuración de una política de bucket. Para obtener información, consulte Exigir MFA. Después de solicitar la MFA para acceder a sus recursos de Amazon S3, la única manera de acceder a estos recursos es proporcionando credenciales temporales que se crean con una clave de MFA. Para obtener más información, consulte la página de detalles de Autenticación multifactor de AWS y Configuración del acceso a una API protegida por MFA en la Guía del usuario de IAM.

Puntos de enlace de solicitud

Usted envía solicitudes REST al punto de enlace predefinido del servicio. Para ver una lista de todos los servicios de AWS y los puntos conexión correspondientes, consulte Regiones y puntos de conexión en la Referencia general de AWS.

Realización de solicitudes a Amazon S3 mediante IPv6

Amazon Simple Storage Service (Amazon S3) admite la capacidad de acceder a los buckets de S3 mediante el Protocolo de Internet, versión 6 (IPv6), además del protocolo IPv4. Los puntos de enlace de doble pila de Amazon S3; admiten solicitudes a buckets de S3 a través de IPv6 y de IPv4. No hay cargos adicionales por acceder a Amazon S3 mediante IPv6. Para obtener más información acerca de los precios, consulte Precios de Amazon S3.

Temas

- Introducción a la realización de solicitudes en IPv6
- Uso de direcciones IPv6 en políticas de IAM

- Probar la compatibilidad de dirección IP
- Uso de puntos de conexión de doble pila en Amazon S3

Introducción a la realización de solicitudes en IPv6

Para hacer una solicitud a un bucket de S3 mediante IPv6, debe utilizar un punto de enlace de doble pila. En la siguiente sección se describe cómo hacer solicitudes mediante IPv6 con los puntos de enlace de doble pila.

A continuación se describen algunos puntos que debe tener en cuenta antes de intentar acceder a un bucket mediante IPv6:

- El cliente y la red que acceden al bucket deben estar autorizados para utilizar IPv6.
- Se admiten tanto solicitudes de estilo alojamiento virtual como de tipo ruta para el acceso a IPv6. Para obtener más información, consulte Puntos de conexión de doble pila en Amazon S3.
- Si utiliza el filtrado de dirección IP de origen en sus políticas de bucket o de usuario de AWS Identity and Access Management (IAM), debe actualizar las políticas para incluir los rangos de direcciones IPv6. Para obtener más información, consulte Uso de direcciones IPv6 en políticas de IAM.
- Cuando utiliza IPv6, los archivos de registro de acceso al servidor producen direcciones IP en un formato de IPv6. Debe actualizar el software, las herramientas y los scripts existentes que utiliza para analizar archivos de registro de Amazon S3 para que puedan analizar las direcciones `Remote IP` con formato de IPv6. Para obtener más información, consulte Formato de registro de acceso al servidor de Amazon S3 y Registro de solicitudes con registro de acceso al servidor.

> **Note**
>
> Si experimenta problemas relacionados con la presencia de direcciones IPv6 en archivos de registro, contacte con AWS Support.

Realización de solicitudes mediante IPv6 con puntos de enlace de doble pila

Puede realizar solicitudes con llamadas a la API de Amazon S3 mediante IPv6 con los puntos de enlace de doble pila. Las operaciones de la API de Amazon S3 funcionan de la misma manera que cuando accede a Amazon S3 mediante IPv6 o IPv4. El desempeño también debe ser el mismo.

Cuando utiliza la API REST, accede directamente al punto de enlace de doble stack. Para obtener más información, consulte Puntos de enlace de doble pila.

Al usar AWS Command Line Interface (AWS CLI) y los SDK de AWS, puede utilizar un parámetro o una marca para cambiar a un punto de enlace de doble pila. También puede especificar el punto de enlace de doble pila directamente como una anulación del punto de enlace de Amazon S3 en el archivo de configuración.

Puede utilizar el punto de enlace de doble pila para acceder a un bucket mediante IPv6 desde cualquiera de las siguientes opciones:

- La AWS CLI, consulte Usar puntos de enlace de doble pila desde la AWS CLI.
- Para los SDK de AWS, consulte Uso de los puntos de enlace de doble pila de los SDK de AWS.
- La API REST, consulte Realizar solicitudes a los puntos de enlace de doble pila con la API REST.

Características no disponibles con IPv6

Actualmente no se admite la siguiente característica cuando se accede a un bucket de S3 a través de IPv6: alojamiento de sitios web estáticos desde un bucket de S3.

Uso de direcciones IPv6 en políticas de IAM

Antes de intentar acceder a un bucket mediante IPv6, debe asegurarse de que las políticas de usuario de IAM o de bucket de S3 utilizadas para el filtrado de direcciones IP estén actualizadas para incluir los rangos de direcciones IPv6. Si no se actualizan la políticas de filtrado de direcciones IP para gestionar direcciones IPv6, los clientes pueden perder u obtener incorrectamente el acceso al bucket cuando comienzan a utilizar IPv6. Para obtener más información acerca de cómo administrar los permisos de acceso con IAM, consulte Administración de identidades y accesos para Amazon S3.

Las políticas de IAM que filtran direcciones IP utilizan operadores de condición de dirección IP. La siguiente política de buckets identifica el rango 54.240.143.* de direcciones IPv4 permitidas mediante operadores de condición de dirección IP. Cualquier dirección IP fuera de este rango no podrá acceder al bucket (`examplebucket`). Dado que todas las direcciones IPv6 están fuera del rango permitido, esta política evita que las direcciones IPv6 puedan acceder a `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"}
      }
    }
  ]
}
```

Puede modificar el elemento `Condition` de la política de bucket para permitir los rangos de direcciones IPv4 (`54.240.143.0/24`) e IPv6 (`2001:DB8:1234:5678::/64`) como se muestra en el siguiente ejemplo. Puede utilizar el mismo tipo de bloque `Condition` que se muestra en el ejemplo para actualizar las políticas de bucket y de usuario de IAM.

```
      "Condition": {
        "IpAddress": {
           "aws:SourceIp": [
             "54.240.143.0/24",
              "2001:DB8:1234:5678::/64"
            ]
         }
       }
```

Antes de utilizar IPv6, debe actualizar todas las políticas de bucket y de usuario de IAM; relevantes que utilizan filtrado de direcciones IP. No recomendamos utilizar el filtrado de direcciones IP en políticas de bucket.

Puede revisar sus políticas de usuario de IAM en la consola de IAM en https://console.aws.amazon.com/iam/. Para obtener más información acerca de IAM, consulte la guía del usuario de IAM. Para obtener información sobre las políticas de buckets de S3, consulte Agregar una política de bucket mediante la consola de Amazon S3.

Probar la compatibilidad de dirección IP

Si utiliza Linux/Unix o Mac OS X, puede probar si tiene acceso a un punto de enlace de doble pila mediante IPv6 con el comando `curl` como se muestra en el siguiente ejemplo:

Example

```
curl -v  http://s3.dualstack.us-west-2.amazonaws.com/
```

Usted obtiene información similar a la del siguiente ejemplo. Si está conectado mediante IPv6, la dirección IP conectada será una dirección 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 utiliza Microsoft Windows 7 o Windows 10, puede probar si tiene acceso a un punto de enlace de doble pila mediante IPv6 o IPv4 con el comando `ping` como se muestra en el siguiente ejemplo.

```
ping ipv6.s3.dualstack.us-west-2.amazonaws.com
```

Uso de puntos de conexión de doble pila en Amazon S3

Los puntos de enlace de doble pila de Amazon S3; admiten solicitudes a buckets de S3 a través de IPv6 y de IPv4. En esta sección se describe cómo utilizar los puntos de enlace de doble pila.

Temas

- Puntos de conexión de doble pila en Amazon S3
- Usar puntos de enlace de doble pila desde la AWS CLI
- Uso de los puntos de enlace de doble pila de los SDK de AWS
- Usar los puntos de enlace de doble pila desde la API REST

Puntos de conexión de doble pila en Amazon S3

Cuando realiza una solicitud a un punto de enlace de doble pila, la URL del bucket resuelve a una dirección IPv6 o IPv4. Para obtener más información acerca de cómo obtener acceso a un bucket mediante IPv6, consulte Realización de solicitudes a Amazon S3 mediante IPv6.

Cuando utiliza la API REST, accede directamente a un punto de enlace de Amazon S3 mediante el nombre del punto de enlace (URI). Puede obtener acceso a un bucket de S3 mediante un punto de conexión de doble pila con un nombre de punto de conexión de estilo alojado virtual o de estilo ruta. Amazon S3 solo es compatible con nombres de puntos de enlace de doble pila regionales, lo que implica que debe especificar la región como parte del nombre.

Use las siguientes convenciones de nomenclatura para los nombres de punto de conexión de doble pila de estilo alojado virtual y de estilo de ruta

- punto de conexión de doble pila de estilo alojado virtual:

  bucketname.s3.dualstack.*aws-region*.amazonaws.com

- punto de conexión de doble pila de estilo de ruta:

  s3.dualstack.*aws-region*.amazonaws.com/*bucketname*

Para obtener más información acerca de los estilos de denominación de los puntos de enlace, consulte Acceso y publicación de un bucket de Amazon S3. Para ver una lista de los puntos de conexión de Amazon S3, consulte Regiones y puntos de conexión en la Referencia general de AWS.

> **Important**
>
> Puede usar transfer acceleration con los puntos de enlace de doble pila. Para obtener más información, consulte Introducción a Amazon S3 Transfer Acceleration.

> **Note**
>
> Los dos tipos de puntos de conexión de VPC para acceder a Amazon S3 (Puntos de conexión de VPC de tipo interfaz y Puntos de conexión de VPC de puerta de enlace) no son compatibles con la doble pila. Para obtener más información acerca de los puntos de conexión de VPC de Amazon S3, consulte AWS PrivateLink para Amazon S3.

Al usar AWS Command Line Interface (AWS CLI) y los SDK de AWS, puede utilizar un parámetro o una marca para cambiar a un punto de enlace de doble pila. También puede especificar el punto

de enlace de doble pila directamente como una anulación del punto de enlace de Amazon S3 en el archivo de configuración. En las secciones siguientes, se describe cómo utilizar los puntos de enlace de doble pila desde la AWS CLI y los SDK de AWS.

Usar puntos de enlace de doble pila desde la AWS CLI

Esta sección proporciona ejemplos de comandos de la AWS CLI. que se usan para realizar solicitudes a un punto de conexión de doble pila. Para obtener instrucciones acerca de cómo configurar la AWS CLI, consulte Desarrollo con Amazon S3 mediante la AWS CLI.

Puede establecer el valor de configuración `use_dualstack_endpoint` a `true` en un perfil de su archivo de AWS Config para dirigir todas las solicitudes de Amazon S3 que realicen los comandos `s3` y `s3api` de la AWS CLI al punto de enlace de doble pila para la región especificada. Puede especificar la región en el archivo de configuración o en un comando utilizando la opción `--region`.

Cuando utilice puntos de enlace de doble pila con la AWS CLI, puede utilizar los estilos de direccionamiento `path` y `virtual`. El estilo de direccionamiento configurado en el archivo de configuración controla si el nombre del bucket está en el hostname o es parte de la URL. La CLI intentará, de manera predeterminada, usar el estilo virtual siempre que sea posible, pero si es necesario recurrirá al estilo de ruta. Para obtener más información, consulte Configuración de la AWS CLI de Amazon S3).

También puede realizar cambios a la configuración mediante un comando, como se muestra en el siguiente ejemplo, en el que se configura `use_dualstack_endpoint` como `true` y `addressing_style` como `virtual` en el perfil predeterminado.

```
$ aws configure set default.s3.use_dualstack_endpoint true
$ aws configure set default.s3.addressing_style virtual
```

Si quiere utilizar un punto de conexión de doble pila exclusivamente para comandos de la AWS CLI específicos (no para todos los comandos), puede usar uno de los métodos siguientes:

- Puede utilizar el punto de conexión de doble pila por cada comando configurando el parámetro `--endpoint-url` como `https://s3.dualstack.`*`aws-region`*`.amazonaws.com` o `http://s3.dualstack.`*`aws-region`*`.amazonaws.com` para cualquier comando `s3` o `s3api`.

  ```
  $ aws s3api list-objects --bucket bucketname --endpoint-url https://s3.dualstack.aws-region.amazonaws.com
  ```

- Puede configurar perfiles separados en su archivo de AWS Config. Por ejemplo, puede crear un perfil que configure `use_dualstack_endpoint` como `true` y un perfil que no configure `use_dualstack_endpoint`. Al ejecutar un comando, especifique qué perfil quiere usar, en función de si quiere usar el punto de conexión de doble pila o no.

> **Note**
>
> Por el momento, al usar la AWS CLI no podrá utilizar la transfer acceleration con puntos de enlace de doble pila. Sin embargo, pronto ofreceremos compatibilidad con la AWS CLI. Para obtener más información, consulte Uso de la AWS CLI.

Uso de los puntos de enlace de doble pila de los SDK de AWS

En esta sección, se proporcionan ejemplos de cómo obtener acceso a un punto de enlace de doble pila con los SDK de AWS.

AWS SDK for JavaEjemplo de punto de enlace de doble pila con

En el siguiente ejemplo, se muestra cómo habilitar puntos de enlace de doble pila al crear un cliente de Amazon S3 mediante AWS SDK for Java.

Para obtener instrucciones sobre cómo crear y probar una muestra funcional de Java, consulte Introducción en la Guía del desarrollador de AWS SDK for Java.

```
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 está utilizando AWS SDK for Java en Windows, puede que tenga que configurar adecuadamente la siguiente propiedad de la máquina virtual Java (JVM):

```
java.net.preferIPv6Addresses=true
```

AWSEjemplo de punto de enlace de doble pila con el SDK para .NET

Cuando utiliza el SDK de AWS para .NET, puede usar la clase `AmazonS3Config` para habilitar el uso de un punto de enlace de doble pila, como se muestra en el siguiente ejemplo.

```
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());
            }
        }
    }
}
```

Para ver una muestra completa en .NET para enumerar objetos, consulte Descripción de claves de objeto mediante programación.

Para obtener información acerca de cómo configurar y ejecutar ejemplos de código, consulte Introducción al SDK de AWS para .NET en la Guía para desarrolladores del SDK de AWS para .NET.

Usar los puntos de enlace de doble pila desde la API REST

Para obtener información sobre cómo realizar solicitudes a puntos de enlace de doble pila con la API REST, consulte Realizar solicitudes a los puntos de enlace de doble pila con la API REST.

Realización de solicitudes con los SDK de AWS

Temas

Puede enviar solicitudes autenticadas a Amazon S3 con los SDK de AWS o mediante las llamadas a la API de REST directamente en su aplicación. La API del SDK de AWS utiliza las credenciales que proporciona para computar la firma para autenticación. Si usa la API de REST directamente en su aplicación, debe escribir el código necesario para computar la firma a fin de autenticar su solicitud. Para ver una lista de los SDK de AWS disponibles, consulte Código de muestra y bibliotecas.

Realización de solicitudes con las credenciales de usuario de IAM o Cuenta de AWS

Puede usar sus credenciales de seguridad de usuario de IAM o su Cuenta de AWS para enviar solicitudes autenticadas a Amazon S3. En esta sección se proporcionan ejemplos de cómo enviar solicitudes autenticadas con AWS SDK for Java, AWS SDK for .NET y AWS SDK for PHP. Para ver una lista de los SDK de AWS disponibles, consulte Código de muestra y bibliotecas.

Cada uno de estos SDK de AWS utiliza una cadena de proveedores de credenciales específica del SDK para encontrar y utilizar las credenciales, y realizar acciones en nombre del propietario de las credenciales. Todas estas cadenas de proveedores de credenciales tienen en común que todas buscan el archivo de credenciales local de AWS.

Para obtener más información, consulte los siguientes temas:

Temas

Para crear un archivo local de credenciales de AWS

La forma más sencilla de configurar credenciales para los SDK de AWS es utilizar un archivo de credenciales de AWS. Si utiliza AWS Command Line Interface (AWS CLI), es posible que ya

tenga configurado un archivo local de credenciales de AWS. De lo contrario, utilice el siguiente procedimiento para configurar un archivo de credenciales:

1. Inicie sesión en la AWS Management Console y abra la consola de IAM en https://console.aws.amazon.com/iam/.
2. Cree un usuario con permisos limitados a los servicios y acciones a los que desea que tenga acceso el código. Para obtener más información acerca de cómo crear un usuario nuevo, consulte Creación de usuarios de IAM (consola) y siga las instrucciones a través del paso 8.
3. Elija Download .csv (Descargar .csv) para guardar una copia local de sus credenciales de AWS.
4. En su equipo, vaya al directorio principal y cree un directorio `.aws`. En sistemas basados en Unix, como Linux u OS X, se encuentra en la siguiente ubicación:

   ```
   ~/.aws
   ```

   En Windows, se encuentra en la siguiente ubicación:

   ```
   %HOMEPATH%\.aws
   ```
5. En el directorio `.aws`, cree un archivo denominado `credentials`.
6. Abra el archivo `.csv` de credenciales que descargó de la consola de IAM y copie su contenido en el archivo de `credentials` con el siguiente formato:

   ```
   [default]
   aws_access_key_id = your_access_key_id
   aws_secret_access_key = your_secret_access_key
   ```
7. Guarde el archivo `credentials` y elimine el archivo `.csv` que descargó en el paso 3.

El archivo de credenciales compartidas ahora está configurado en el equipo local y está listo para su uso con los SDK de AWS.

Envío de solicitudes autenticadas mediante los SDK de AWS

Utilice los SDK de AWS para enviar solicitudes autenticadas. Para obtener más información sobre el envío de solicitudes autenticadas, consulte Credenciales de seguridad de AWS o IAM Identity Center Authentication (Autenticación del Centro de identidades de IAM).

Java

Para enviar solicitudes autenticadas a Amazon S3 mediante sus credenciales de usuario de IAM o su Cuenta de AWS, haga lo siguiente:

- Use la clase `AmazonS3ClientBuilder` para crear una instancia `AmazonS3Client`.
- Ejecute uno de los métodos de `AmazonS3Client` para enviar solicitudes a Amazon S3. El cliente generará la firma necesaria a partir de las credenciales que proporcione y la incluirá en la solicitud.

En el siguiente ejemplo se realizan las tareas anteriores. Para obtener información sobre cómo crear y probar una muestra funcional, consulte Introducción en la Guía del desarrollador de AWS SDK for Java.

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

Para enviar solicitudes autenticadas mediante sus credenciales de usuario de IAM o su Cuenta de AWS:

- Cree una instancia de la clase `AmazonS3Client`.
- Ejecute uno de los métodos de `AmazonS3Client` para enviar solicitudes a Amazon S3. El cliente generará la firma necesaria a partir de las credenciales que se le proporcionan y la incluirá en la solicitud que envíe a Amazon S3.

Para obtener más información, consulte Realización de solicitudes con las credenciales de usuario de IAM o Cuenta de AWS.

> **Note**
>
> - Puede crear el cliente `AmazonS3Client` sin facilitar sus credenciales de seguridad. Las solicitudes enviadas con este cliente son solicitudes anónimas y no tienen firma. Amazon S3 devuelve un error si envía solicitudes anónimas para un recurso que no esté disponible públicamente.
> - Puede crear una Cuenta de AWS y crear los usuarios necesarios. También puede administrar las credenciales para esos usuarios. Necesita estas credenciales para realizar la tarea del siguiente ejemplo. Para obtener más información, consulte Configuración de credenciales de AWS en la Guía para desarrolladores de AWS SDK for .NET.
>
>   A continuación, también puede configurar la aplicación para recuperar perfiles y credenciales de forma activa y luego usar explícitamente dichas credenciales al crear un cliente del servicio de AWS. Para obtener más información, consulte Acceso a credenciales y perfiles en una aplicación en la Guía para desarrolladores de AWS SDK for .NET.

En el siguiente ejemplo de código C# se muestra cómo realizar las tareas anteriores. Para obtener información acerca de cómo configurar y ejecutar ejemplos de código, consulte Introducción al SDK de AWS para .NET en la Guía para desarrolladores del SDK de AWS para .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);
            }
        }
    }
}
```

Para ver ejemplos funcionales, consulte Información general de los objetos de Amazon S3 y Descripción general de los buckets. Puede probar estos ejemplos con credenciales de usuario de IAM o su Cuenta de AWS.

Por ejemplo, para enumerar todas las claves de objetos de su bucket, consulte Descripción de claves de objeto mediante programación.

PHP

En esta sección, se explica el uso de una clase de la versión 3 de AWS SDK for PHP para enviar solicitudes autenticadas con las credenciales de usuario de IAM o su Cuenta de AWS. Para obtener más información acerca de la API del SDK de AWS para Ruby, consulte SDK de AWS para Ruby, versión 2.

El siguiente ejemplo de PHP muestra cómo el cliente realiza una solicitud con las credenciales de seguridad para enumerar todos los buckets de la cuenta.

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**
>
> Puede crear el cliente `S3Client` sin facilitar sus credenciales de seguridad. Las solicitudes enviadas con este cliente son solicitudes anónimas y no tienen firma. Amazon S3 devuelve un error si envía solicitudes anónimas para un recurso que no esté disponible públicamente. Para obtener más información, consulte Creación de clientes anónimos en la Documentación de AWS SDK for PHP.

Para ver ejemplos funcionales, consulte Información general de los objetos de Amazon S3. Puede probar estos ejemplos con credenciales de usuario de IAM o su Cuenta de AWS.

Para ver un ejemplo de enumeración de claves de objetos en un bucket, consulte Descripción de claves de objeto mediante programación.

Ruby

A fin de poder utilizar la versión 3 de AWS SDK for Ruby para hacer llamadas a Amazon S3, debe establecer las credenciales de acceso de AWS que utiliza el SDK para comprobar el acceso a los buckets y los objetos. Si ha configurado las credenciales compartidas en el perfil de credenciales de AWS de su sistema local, la versión 3 del SDK para Ruby puede utilizarlas, lo que le evitará tener que declararlas en el código. Para obtener más información acerca de cómo configurar las credenciales compartidas, consulte Realización de solicitudes con las credenciales de usuario de IAM o Cuenta de AWS.

En el siguiente fragmento de código de Ruby, se utilizan las credenciales de un archivo de credenciales de AWS compartido en un equipo local para autenticar una solicitud para obtener todos los nombres de claves de objeto de un bucket específico. Hace lo siguiente:

1. Crea una instancia de la clase `Aws::S3::Client`.
2. Realiza una solicitud a Amazon S3 enumerando los objetos de un bucket con el método `list_objects_v2` de `Aws::S3::Client`. El cliente genera el valor de firma necesario a partir de las credenciales del archivo de credenciales de AWS del equipo y lo incluye en la solicitud que envía a Amazon S3.
3. Imprime la matriz de nombres de claves de objeto en el 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__
```

Aunque no tenga un archivo de credenciales de AWS local, puede crear el recurso `Aws::S3::Client` y ejecutar código en los buckets y los objetos de Amazon S3. Las solicitudes que se envían utilizando la versión 3 del SDK para Ruby son anónimas; de forma predeterminada, no tienen firma. Amazon S3 devuelve un error si envía solicitudes anónimas para un recurso que no esté disponible públicamente.

Puede utilizar y ampliar el fragmento de código anterior para aplicaciones del SDK para Ruby como se muestra en el siguiente ejemplo, que es más robusto.

```
# 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

En el siguiente ejemplo, se utilizan credenciales de AWS cargadas automáticamente por el SDK para Go desde el archivo de credenciales compartido.

```
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)
  }
 }
}
```

Recursos relacionados

- Desarrollo con Amazon S3 mediante los SDK de AWS
- AWS SDK for PHP para la clase Aws\S3\S3Client de Amazon S3
- Documentación de AWS SDK for PHP

Realización de solicitudes con las credenciales temporales de usuario de IAM

Una Cuenta de AWS o un usuario de IAM puede solicitar las credenciales de seguridad temporales y utilizarlas para enviar solicitudes autenticadas a Amazon S3. En esta sección, se proporcionan ejemplos sobre cómo utilizar AWS SDK for Java, .NET y PHP para obtener las credenciales de seguridad temporales y utilizarlas con el fin de autenticar las solicitudes para Amazon S3.

Java

Un usuario de IAM o una Cuenta de AWS pueden solicitar credenciales de seguridad temporales (consulte Realizar solicitudes) mediante AWS SDK for Java y utilizarlas para obtener acceso a Amazon S3. Estas credenciales caducan cuando termine la sesión especificada.

De forma predeterminada, la sesión durará una hora. Si utiliza credenciales de usuario de IAM, puede especificar la duración al solicitar las credenciales de seguridad temporales desde 15 minutos hasta la duración máxima de sesión del rol. Para obtener más información acerca de las credenciales de seguridad temporales, consulte Credenciales de seguridad temporales en la guía del usuario de IAM. Para obtener más información acerca de la realización de solicitudes, consulte Realizar solicitudes.

Para obtener credenciales de seguridad temporales y acceso a Amazon S3

1. Cree una instancia de la clase `AWSSecurityTokenService`. Para obtener información acerca de cómo proporcionar las credenciales, consulte Desarrollo con Amazon S3 mediante los SDK de AWS.
2. Recupere las credenciales de seguridad temporales para el rol deseado al llamar al método `assumeRole()` del cliente de Security Token Service (STS).
3. Empaquete las credenciales de seguridad temporales en un objeto `BasicSessionCredentials`. Utilice este objeto para proporcionar las credenciales de seguridad temporales al cliente de Amazon S3.
4. Cree una instancia de la clase `AmazonS3Client` usando las credenciales de seguridad temporales. Envíe las solicitudes a Amazon S3 con este cliente. Si envía las solicitudes con las credenciales vencidas, Amazon S3 devolverá un error.

> **Note**
>
> Si obtiene las credenciales de seguridad temporales con las credenciales de seguridad de su Cuenta de AWS, las credenciales de seguridad temporales tendrán validez durante

una hora únicamente. Puede especificar la duración de la sesión solo si utiliza las credenciales del usuario de IAM para solicitar una sesión.

En el siguiente ejemplo se muestra un conjunto de claves de objeto del bucket especificado. En el ejemplo se obtienen credenciales de seguridad temporales para una sesión y se utilizan para enviar una solicitud autenticada a Amazon S3.

Si desea probar el ejemplo con las credenciales de usuario de IAM, debe crear un usuario de IAM en la Cuenta de AWS. Para obtener más información acerca de cómo crear un usuario de IAM, consulte Creación del primer grupo de usuarios y administradores de IAM en la Guía del usuario de IAM.

Para obtener instrucciones sobre cómo crear y probar una muestra funcional, consulte Introducción en la Guía del desarrollador de AWS SDK for Java.

```
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 usuario de IAM o una Cuenta de AWS pueden solicitar credenciales de seguridad temporales mediante AWS SDK for .NET y utilizarlas para obtener acceso a Amazon S3. Estas credenciales caducan cuando termine la sesión.

De forma predeterminada, la sesión durará una hora. Si utiliza credenciales de usuario de IAM, puede especificar la duración al solicitar las credenciales de seguridad temporales desde 15 minutos hasta la duración máxima de sesión del rol. Para obtener más información acerca de las credenciales de seguridad temporales, consulte Credenciales de seguridad temporales en la guía del usuario de IAM. Para obtener más información acerca de la realización de solicitudes, consulte Realizar solicitudes.

Para obtener credenciales de seguridad temporales y acceso a Amazon S3

1. Cree una instancia del cliente de AWS Security Token Service. `AmazonSecurityTokenServiceClient`. Para obtener información acerca de cómo proporcionar las credenciales, consulte Desarrollo con Amazon S3 mediante los SDK de AWS.
2. Inicie una sesión utilizando el método `GetSessionToken` del cliente de STS que creó en el paso anterior. Proporcione información acerca de la sesión para este método con un objeto `GetSessionTokenRequest`.

   El método devuelve las credenciales de seguridad temporales.
3. Empaquete las credenciales de seguridad temporales en una instancia del objeto `SessionAWSCredentials`. Utilice este objeto para proporcionar las credenciales de seguridad temporales al cliente de Amazon S3.

4. Cree una instancia de la clase `AmazonS3Client` al proporcionar las credenciales de seguridad temporales. Envíe las solicitudes a Amazon S3 con este cliente. Si envía las solicitudes con las credenciales vencidas, Amazon S3 devolverá un error.

> **Note**
>
> Si obtiene las credenciales de seguridad temporales con las credenciales de seguridad de su Cuenta de AWS, esas credenciales tendrán validez durante una hora únicamente. Puede especificar la duración de una sesión solo si utiliza las credenciales de usuario de IAM para solicitar una sesión.

En el siguiente ejemplo de código C# se muestran las claves de objeto del bucket especificado. A modo de ilustración, en el ejemplo se obtienen las credenciales de seguridad temporales para una sesión predeterminada de una hora y se utilizan para enviar una solicitud autenticada a Amazon S3.

Si desea probar el ejemplo con las credenciales de usuario de IAM, debe crear un usuario de IAM en la Cuenta de AWS. Para obtener más información acerca de cómo crear un usuario de IAM, consulte Creación del primer grupo de usuarios y administradores de IAM en la Guía del usuario de IAM. Para obtener más información acerca de la realización de solicitudes, consulte Realizar solicitudes.

Para obtener información acerca de cómo configurar y ejecutar ejemplos de código, consulte Introducción al SDK de AWS para .NET en la Guía para desarrolladores del SDK de AWS para .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

Para obtener más información acerca de la API del SDK de AWS para Ruby, consulte SDK de AWS para Ruby, versión 2.

Tanto un usuario de IAM como una Cuenta de AWS pueden solicitar credenciales de seguridad temporales mediante la versión 3 de AWS SDK for PHP. A continuación, puede usar las credenciales temporales para acceder a Amazon S3. Las credenciales caducan cuando finaliza la duración de la sesión.

De forma predeterminada, la sesión durará una hora. Si utiliza credenciales de usuario de IAM, puede especificar la duración al solicitar las credenciales de seguridad temporales desde 15

minutos hasta la duración máxima de sesión del rol. Para obtener más información acerca de las credenciales de seguridad temporales, consulte Credenciales de seguridad temporales en la guía del usuario de IAM. Para obtener más información acerca de la realización de solicitudes, consulte Realizar solicitudes.

> **Note**
>
> Si obtiene las credenciales de seguridad temporales con las credenciales de seguridad de su Cuenta de AWS, estas tendrán validez durante una hora únicamente. Puede especificar la duración de la sesión solo si utiliza las credenciales del usuario de IAM para solicitar una sesión.

Example

En el siguiente ejemplo de PHP se muestran las claves de objeto del bucket especificado con las credenciales de seguridad temporales. En el ejemplo se obtienen credenciales de seguridad temporales para una sesión predeterminada de una hora y se utilizan para enviar una solicitud autenticada a Amazon S3. Para obtener más información acerca de la API del SDK de AWS para Ruby, consulte SDK de AWS para Ruby, versión 2.

Si desea probar el ejemplo con las credenciales de usuario de IAM, debe crear un usuario de IAM en la Cuenta de AWS. Para obtener información acerca de cómo crear un usuario de IAM, consulte Creación del primer grupo de usuarios y administradores de IAM en la Guía del usuario de IAM. Para obtener ejemplos de configuración de la duración de la sesión con las credenciales de usuario de IAM para solicitar una sesión, consulte Realización de solicitudes con las credenciales temporales de usuario de 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 usuario de IAM o una Cuenta de AWS pueden solicitar credenciales de seguridad temporales mediante AWS SDK for Ruby y utilizarlas para obtener acceso a Amazon S3. Estas credenciales caducan cuando termine la sesión.

De forma predeterminada, la sesión durará una hora. Si utiliza credenciales de usuario de IAM, puede especificar la duración al solicitar las credenciales de seguridad temporales desde 15 minutos hasta la duración máxima de sesión del rol. Para obtener más información acerca de las credenciales de seguridad temporales, consulte Credenciales de seguridad temporales en la

guía del usuario de IAM. Para obtener más información acerca de la realización de solicitudes, consulte Realizar solicitudes.

> **Note**
>
> Si obtiene las credenciales de seguridad temporales con las credenciales de seguridad de su Cuenta de AWS, estas tendrán validez durante una hora únicamente. Puede especificar la duración de la sesión solo si utiliza las credenciales del usuario de IAM para solicitar una sesión.

El siguiente ejemplo de código Ruby crea un usuario temporal para mostrar los elementos en un bucket especificado durante una hora. Para utilizar este ejemplo, debe tener credenciales de AWS con los permisos necesarios para crear nuevos clientes de AWS Security Token Service (AWS STS) y mostrar buckets de 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
```

Recursos relacionados

- Desarrollo con Amazon S3 mediante los SDK de AWS
- AWS SDK for PHP para la clase Aws\S3\S3Client de Amazon S3
- Documentación de AWS SDK for PHP

Realizar solicitudes con credenciales temporales de usuario federado

Puede solicitar credenciales de seguridad temporales y proporcionarlas a sus usuarios federados o aplicaciones que necesitan acceso a sus recursos de AWS. En esta sección, se proporcionan ejemplos de cómo puede utilizar el SDK de AWS para obtener credenciales de seguridad temporales para sus usuarios federados o aplicaciones y enviar solicitudes autenticadas a Amazon S3 con esas credenciales. Para ver una lista de los SDK de AWS disponibles, consulte Código de muestra y bibliotecas.

> **Note**
>
> Tanto la Cuenta de AWS como un usuario de IAM pueden solicitar credenciales de seguridad temporales para usuarios federados. Sin embargo, para mayor seguridad, solo un usuario de IAM con los permisos necesarios debe solicitar estas credenciales temporales para asegurarse de que el usuario federado obtenga al menos los permisos del usuario de IAM que realiza la solicitud. En algunas aplicaciones, puede resultarle útil crear un usuario de IAM con permisos específicos con el único fin de proporcionar credenciales de seguridad temporales a sus usuarios federados y aplicaciones.

Java

Puede proporcionar credenciales de seguridad temporales para sus usuarios federados y aplicaciones, de modo que estos puedan enviar solicitudes autenticadas a fin de obtener acceso a sus recursos de AWS. Al solicitar estas credenciales temporales, debe proporcionar un nombre de usuario y una política de IAM que describa los permisos de recursos que desea otorgar. De forma predeterminada, la sesión durará una hora. Puede solicitar las credenciales de seguridad temporales para usuarios federados y aplicaciones para establecer un valor de duración distinto de forma explícita.

> **Note**
>
> Para mayor seguridad, cuando solicite las credenciales de seguridad temporales para usuarios federados y aplicaciones, recomendamos que use un usuario de IAM específico que solo tenga los permisos de acceso necesarios. El usuario temporal creado nunca puede obtener más permisos que el usuario de IAM que solicitó las credenciales de seguridad temporales. Para obtener más información, consulte Preguntas frecuentes de AWS Identity and Access Management.

Para proporcionar credenciales de seguridad y utilizar una solicitud autenticada para obtener acceso a recursos, haga lo siguiente:

- Cree una instancia de la clase `AWSSecurityTokenServiceClient`.
- Inicie una sesión utilizando el método `getFederationToken()` del cliente Security Token Service (STS). Proporcione información acerca de la sesión, como el nombre de usuario y una política de IAM que desee adjuntar a las credenciales temporales. Puede proporcionar una duración de sesión opcional. Este método devuelve sus credenciales de seguridad temporales.
- Empaquete las credenciales de seguridad temporales en una instancia del objeto `BasicSessionCredentials`. Utilice este objeto para proporcionar las credenciales de seguridad temporales al cliente de Amazon S3.
- Cree una instancia de la clase `AmazonS3Client` usando las credenciales de seguridad temporales. Envíe las solicitudes a Amazon S3 con este cliente. Si envía las solicitudes con las credenciales vencidas, Amazon S3 devolverá un error.

Example

En el siguiente ejemplo se enumeran las claves del bucket de S3 especificado. En el ejemplo obtiene las credenciales de seguridad temporales para una sesión de dos horas para su usuario federado y utiliza las credenciales para enviar solicitudes autenticadas a Amazon S3. Para ejecutar el ejemplo, debe crear un usuario de IAM con una política adjuntada que permita al usuario solicitar credenciales de seguridad temporales y generar una lista de sus recursos de AWS. La siguiente política lo cumple:

```
{
  "Statement":[{
      "Action":["s3:ListBucket",
        "sts:GetFederationToken*"
      ],
      "Effect":"Allow",
      "Resource":"*"
    }
  ]
}
```

Para obtener más información acerca de cómo crear un usuario de IAM, consulte Creación del primer grupo de usuarios y administradores de IAM en la Guía del usuario de IAM.

Después de crear un usuario de IAM y asociarle la política anterior, puede ejecutar el ejemplo siguiente. Para obtener instrucciones sobre cómo crear y probar una muestra funcional, consulte Introducción en la Guía del desarrollador de AWS SDK for Java.

```
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

Puede proporcionar credenciales de seguridad temporales para sus usuarios federados y aplicaciones, de modo que estos puedan enviar solicitudes autenticadas a fin de obtener acceso a sus recursos de AWS. Al solicitar estas credenciales temporales, debe proporcionar un nombre de usuario y una política de IAM que describa los permisos de recursos que desea otorgar. De forma predeterminada, la sesión dura una hora. Puede solicitar las credenciales de seguridad temporales para usuarios federados y aplicaciones para establecer un valor de duración distinto de forma explícita. Para obtener información acerca del envío de solicitudes autenticadas, consulte Realizar solicitudes.

> **Note**
>
> Cuando solicite las credenciales de seguridad temporales para usuarios federados y aplicaciones, para mayor seguridad le sugerimos que use un usuario de IAM específico que solo tenga los permisos de acceso necesarios. El usuario temporal creado nunca puede obtener más permisos que el usuario de IAM que solicitó las credenciales de seguridad temporales. Para obtener más información, consulte Preguntas frecuentes de AWS Identity and Access Management.

Siga estas instrucciones:

- Cree una instancia del cliente de AWS Security Token Service, clase `AmazonSecurityTokenServiceClient`.
- Inicie sesión utilizando el método `GetFederationToken` del cliente STS. Tiene que proporcionar información acerca de la sesión, como el nombre de usuario y una política de IAM que desee adjuntar a las credenciales temporales. También tiene la posibilidad de proporcionar una duración de sesión. Este método devuelve sus credenciales de seguridad temporales.
- Empaquete las credenciales de seguridad temporales en una instancia del objeto `SessionAWSCredentials`. Utilice este objeto para proporcionar las credenciales de seguridad temporales al cliente de Amazon S3.

- Cree una instancia de la clase `AmazonS3Client` proporcionando las credenciales de seguridad temporales. Use este cliente para enviar solicitudes a Amazon S3. Si envía las solicitudes con las credenciales vencidas, Amazon S3 devolverá un error.

Example

En el siguiente ejemplo de C# se enumeran las claves del bucket especificado. En el ejemplo se obtienen las credenciales de seguridad temporales para una sesión de dos horas para el usuario federado (User1) y se utilizan las credenciales para enviar solicitudes autenticadas a Amazon S3.

- Para este ejercicio cree un usuario de IAM con permisos mínimos. Con las credenciales de este usuario de IAM, solicite credenciales temporales para otros usuarios. Este ejemplo solo enumera los objetos de un bucket específico. Cree un usuario de IAM con la siguiente política asociada:

```
{
  "Statement":[{
      "Action":["s3:ListBucket",
        "sts:GetFederationToken*"
      ],
      "Effect":"Allow",
      "Resource":"*"
    }
  ]
}
```

  La política permite al usuario de IAM solicitar las credenciales de seguridad temporales y el permiso de acceso solo para mostrar sus recursos de AWS. Para obtener más información acerca de cómo crear un usuario de IAM, consulte Creación del grupo de usuarios y administradores de IAM en la Guía del usuario de IAM.
- Use las credenciales de seguridad del usuario de IAM para probar el siguiente ejemplo. En el ejemplo se envía una solicitud autenticada a Amazon S3 con las credenciales de seguridad temporales. En el ejemplo se especifica la siguiente política cuando se solicitan las credenciales de seguridad temporales para el usuario federado (User1), que restringe el acceso para mostrar los objetos de un bucket específico (`YourBucketName`). Debe actualizar la política y proporcionar su propio nombre de bucket existente.

```
{
  "Statement":[
```

```
    {
      "Sid":"1",
      "Action":["s3:ListBucket"],
      "Effect":"Allow",
      "Resource":"arn:aws:s3:::YourBucketName"
    }
  ]
}
```

- Example

  Actualice la siguiente muestra y proporcione el nombre de bucket que especificó en la política de acceso del usuario federado anterior. Para obtener información acerca de cómo configurar y ejecutar ejemplos de código, consulte Introducción al SDK de AWS para .NET en la Guía para desarrolladores del SDK de AWS para .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

En este tema, se explica cómo usar clases de la versión 3 de AWS SDK for PHP a fin de solicitar credenciales de seguridad temporales para usuarios federados y aplicaciones, y utilizarlas para obtener acceso a recursos almacenados en Amazon S3. Para obtener más información acerca de la API del SDK de AWS para Ruby, consulte SDK de AWS para Ruby, versión 2.

Puede proporcionar credenciales de seguridad temporales a sus usuarios federados y aplicaciones para que puedan enviar solicitudes autenticadas a fin de obtener acceso a sus

recursos de AWS. Al solicitar estas credenciales temporales, debe proporcionar un nombre de usuario y una política de IAM que describa los permisos de recursos que desea otorgar. Estas credenciales caducan cuando finaliza la duración de la sesión. De forma predeterminada, la sesión durará una hora. Puede solicitar las credenciales de seguridad temporales para usuarios federados y aplicaciones para establecer explícitamente otro valor para la duración. Para obtener más información acerca de las credenciales de seguridad temporales, consulte Credenciales de seguridad temporales en la guía del usuario de IAM. Para obtener más información sobre cómo proporcionar credenciales de seguridad temporales a sus usuarios federados y sus aplicaciones, consulte Realizar solicitudes.

Para mayor seguridad, cuando solicite las credenciales de seguridad temporales para usuarios federados y aplicaciones, recomendamos que use un usuario de IAM específico que solo tenga los permisos de acceso necesarios. El usuario temporal creado nunca puede obtener más permisos que el usuario de IAM que solicitó las credenciales de seguridad temporales. Para obtener más información sobre la federación de identidad, consulte Preguntas frecuentes de AWS Identity and Access Management.

Para obtener más información acerca de la API del SDK de AWS para Ruby, consulte SDK de AWS para Ruby, versión 2.

Example

En el siguiente ejemplo de PHP se enumeran las claves del bucket especificado. En el ejemplo se obtienen las credenciales de seguridad temporales para una sesión de una hora para el usuario federado (User1). A continuación, use las credenciales de seguridad temporales para enviar solicitudes autenticadas a Amazon S3.

Para mayor seguridad, cuando solicite las credenciales temporales para otros usuarios, utilice las credenciales de seguridad de un usuario de IAM que tenga permisos para solicitar credenciales de seguridad temporales. Para asegurarse de que el usuario de IAM otorga solo los permisos específicos de la aplicación mínimos al usuario federado, también puede limitar los permisos de acceso a este usuario de IAM. En este ejemplo solo se enumera objetos de un bucket específico. Cree un usuario de IAM con la siguiente política asociada:

```
{
  "Statement":[{
      "Action":["s3:ListBucket",
        "sts:GetFederationToken*"
      ],
      "Effect":"Allow",
```

```
      "Resource":"*"
    }
  ]
}
```

La política permite al usuario de IAM solicitar las credenciales de seguridad temporales y el permiso de acceso solo para mostrar sus recursos de AWS. Para obtener más información acerca de cómo crear un usuario de IAM, consulte Creación del primer grupo de usuarios y administradores de IAM en la Guía del usuario de IAM.

Ahora puede utilizar las credenciales de seguridad del usuario de IAM para probar el siguiente ejemplo. En el ejemplo se envía una solicitud autenticada a Amazon S3 con las credenciales de seguridad temporales. Cuando solicite las credenciales de seguridad temporales para el usuario federado (User1), el ejemplo especifica la política siguiente, que restringe el acceso para enumerar los objetos de un bucket específico. Actualice la política con el nombre de su bucket.

```
{
  "Statement":[
    {
      "Sid":"1",
      "Action":["s3:ListBucket"],
      "Effect":"Allow",
      "Resource":"arn:aws:s3:::YourBucketName"
    }
  ]
}
```

En el siguiente ejemplo, al especificar el recurso de política, reemplace `YourBucketName` por el nombre del bucket.

```
 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

Puede proporcionar credenciales de seguridad temporales para sus usuarios federados y aplicaciones, de modo que estos puedan enviar solicitudes autenticadas a fin de obtener

acceso a sus recursos de AWS. Al solicitar estas credenciales temporales al servicio de IAM, debe proporcionar un nombre de usuario y una política de IAM que describa los permisos de recursos que desea conceder. De forma predeterminada, la sesión durará una hora. Sin embargo, si solicita las credenciales temporales con las credenciales de usuario de IAM, puede solicitar las credenciales de seguridad temporales para usuarios federados y aplicaciones para establecer un valor de duración distinto de forma explícita. Para obtener más información sobre cómo proporcionar credenciales de seguridad temporales para sus usuarios federados y sus aplicaciones, consulte Realizar solicitudes.

> **Note**
>
> Para mayor seguridad, cuando solicita las credenciales de seguridad temporales para usuarios federados y aplicaciones, se recomienda utilizar un usuario de IAM específico que solo tenga los permisos de acceso necesarios. El usuario temporal creado nunca puede obtener más permisos que el usuario de IAM que solicitó las credenciales de seguridad temporales. Para obtener más información, consulte Preguntas frecuentes de AWS Identity and Access Management.

Example

En el siguiente ejemplo de código Ruby se permite que un usuario federado con un conjunto limitado de permisos enumere las clave en el bucket especificado.

```
# 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__
```

Recursos relacionados

- Desarrollo con Amazon S3 mediante los SDK de AWS
- AWS SDK for PHP para la clase Aws\S3\S3Client de Amazon S3
- Documentación de AWS SDK for PHP

Realizar solicitudes con la API REST

Esta sección incluye información acerca de cómo realizar solicitudes a los puntos de enlace de Amazon S3 con la API REST. Para ver una lista de los puntos de conexión de Amazon S3, consulte Regiones y puntos de conexión en la Referencia general de AWS.

Crear nombres de host de S3 para solicitudes de la API REST

Los puntos de enlace de Amazon S3 siguen la estructura que se muestra a continuación:

```
s3.Region.amazonaws.com
```

Los puntos de enlace de puntos de acceso de Amazon S3 y los puntos de enlace de doble pila también siguen la estructura estándar:

- Puntos de acceso de Amazon S -`s3-accesspoint.Region.amazonaws.com`
- Doble pila - `s3.dualstack.Region.amazonaws.com`

Para obtener una lista completa de las regiones y los puntos de conexión de Amazon S3, consulte Puntos de conexión y cuotas de Amazon S3 en la Referencia general de Amazon Web Services.

Solicitudes de tipo alojamiento virtual y de tipo ruta

Cuando realiza solicitudes con la API REST, puede utilizar los URI de tipo alojamiento virtual o tipo ruta para los puntos de enlace de Amazon S3. Para obtener más información, consulte Alojamiento virtual de buckets.

Example Solicitud de tipo alojamiento virtual

A continuación, se muestra un ejemplo de una solicitud de tipo alojamiento virtual para eliminar el archivo `puppy.jpg` del bucket denominado `examplebucket` en la región EE. UU. Oeste (Oregón). Para obtener más información acerca de las solicitudes de estilo de alojamiento virtual, consulte Solicitudes de tipo host virtual.

```
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 Solicitud de tipo ruta

A continuación se muestra un ejemplo de una versión tipo ruta de la misma solicitud.

```
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
```

Actualmente, Amazon S3 admite URL tanto de tipo host virtual como de tipo ruta en todas las Regiones de AWS. Sin embargo, las URL de tipo ruta dejarán de usarse en el futuro. Para obtener más información, consulte la siguiente nota Importante.

Para obtener más información acerca de las solicitudes de tipo ruta, consulte Solicitudes de tipo ruta.

> **Important**
>
> Actualización (23 de septiembre de 2020): para garantizar que los clientes tienen el tiempo necesario para pasar a las URL de tipo host virtual, hemos decidido retrasar la obsolescencia de las URL de tipo ruta. Para obtener más información, consulte Amazon S3 Path Deprecation Plan – The Rest of the Story en el Blog de noticias de AWS.

Realizar solicitudes a los puntos de enlace de doble pila con la API REST

Cuando utiliza la API REST, puede acceder directamente al punto de enlace de doble pila con un nombre de punto de enlace de tipo alojamiento virtual o tipo ruta (URI). Todos los nombres de punto de enlace de doble pila de Amazon S3 incluyen la región en el nombre. A diferencia de los puntos de enlace estándar que solo admiten IPv4, tanto los puntos de enlace de tipo alojamiento virtual como los de tipo ruta utilizan nombres de puntos de enlace específicos de la región.

Example Solicitud de punto de enlace de doble pila de tipo alojamiento virtual

Puede utilizar un punto de enlace de tipo alojamiento virtual en su solicitud REST como se muestra en el siguiente ejemplo, que recupera el objeto `puppy.jpg` del bucket denominado `examplebucket` en la región EE. UU. Oeste (Oregón).

```
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 Solicitud de punto de enlace de doble pila de tipo ruta

O bien, puede utilizar un punto de enlace de tipo ruta en su solicitud como se muestra en el siguiente ejemplo.

```
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
```

Para obtener más información acerca de los puntos de enlace de doble pila, consulte Uso de puntos de conexión de doble pila en Amazon S3.

Para obtener más información acerca de cómo realizar solicitudes mediante la API REST, consulte los temas siguientes.

Temas

- Alojamiento virtual de buckets
- Solicitar redireccionamiento y la API de REST

Alojamiento virtual de buckets

El alojamiento virtual es la práctica que consiste en distribuir múltiples sitios web desde un solo servidor web. Una manera de diferenciar sitios en las solicitudes de la API de REST de Amazon S3 es mediante el uso del nombre de host aparente de Request-URI en lugar de solo la parte del nombre de la ruta del URI. Para especificar un bucket, una solicitud REST de Amazon S3 normal utiliza el primer componente delimitado por una barra inclinada de la ruta del URI de la solicitud. En su lugar, puede utilizar el alojamiento virtual de Amazon S3 para hacer referencia a un bucket en una llamada a la API de REST mediante el encabezado `Host` de HTTP. En la práctica, Amazon S3 interpreta que el `Host` significa que es posible acceder a la mayoría de los buckets de manera automática (para tipos limitados de solicitudes) en `https://`*`bucket-name`*`.s3.`*`region-code`*`.amazonaws.com`. Para obtener una lista completa de las regiones y los puntos de conexión de Amazon S3, consulte Puntos de conexión y cuotas de Amazon S3 en la Referencia general de Amazon Web Services.

El alojamiento virtual también tiene otros beneficios. Al nombrar su bucket después de su nombre de dominio registrado y al convertir ese nombre en un alias de Domain Name System (DNS, Sistema de nombres de dominio) para Amazon S3, puede personalizar completamente el URL de sus recursos

de Amazon S3, por ejemplo, `http://my.bucket-name.com/`. También puede publicar en el "directorio raíz" del servidor virtual de su bucket. Esta capacidad puede ser importante ya que varias aplicaciones existentes buscan archivos en esta ubicación estándar. Por ejemplo, `favicon.ico`, `robots.txt` y `crossdomain.xml` se encuentran en la raíz.

> **Important**
>
> Cuando utiliza buckets de tipo host virtual con SSL, el certificado comodín de SSL solo se asocia a los buckets que no contienen puntos (`.`). Para solucionar esta limitación, use HTTP o escriba su propia lógica de verificación de certificado. Para obtener más información, consulte Amazon S3 Path Deprecation Plan en AWS News Blog.

Temas

Solicitudes de tipo ruta

Actualmente, Amazon S3 admite URL tanto de tipo host virtual como de tipo ruta en todas las Regiones de AWS. Sin embargo, las URL de tipo ruta dejarán de usarse en el futuro. Para obtener más información, consulte la siguiente nota Importante.

En Amazon S3, las URL de tipo ruta utilizan el siguiente formato:

```
https://s3.region-code.amazonaws.com/bucket-name/key-name
```

Por ejemplo, si crea un bucket con el nombre `amzn-s3-demo-bucket1` en la región EE. UU. Oeste (Oregón) y quiere acceder al objeto `puppy.jpg` en dicho bucket, puede usar la siguiente URL de tipo ruta:

```
https://s3.us-west-2.amazonaws.com/amzn-s3-demo-bucket1/puppy.jpg
```

> **Important**
>
> Actualización (23 de septiembre de 2020): para garantizar que los clientes tienen el tiempo necesario para pasar a las URL de tipo host virtual, hemos decidido retrasar la obsolescencia de las URL de tipo ruta. Para obtener más información, consulte Amazon S3 Path Deprecation Plan – The Rest of the Story en el Blog de noticias de AWS.

> **Warning**
>
> Cuando alojes contenido de un sitio web al que se acceda desde un navegador web, evita utilizar la URL de tipo ruta, ya que podrían interferir con el mismo modelo de seguridad de origen del navegador. Para alojar el contenido del sitio web, le recomendamos que utilice puntos de conexión de sitios web de S3 o una distribución de CloudFront. Para obtener más información, consulte Puntos de enlace de sitio web e implemente una aplicación de una sola página basada en React en Amazon S3 y CloudFront en Los patrones de AWS Perspective Guidance.

Solicitudes de tipo host virtual

En un URI de tipo alojamiento virtual, el nombre del bucket forma parte del nombre del dominio en el URL.

Las URL de tipo host virtual de Amazon S3 utilizan el siguiente formato:

```
https://bucket-name.s3.region-code.amazonaws.com/key-name
```

En este ejemplo, `amzn-s3-demo-bucket1` es el nombre del bucket, EE.UU Oeste (Oregon) es la región y `puppy.png` es el nombre clave:

```
https://amzn-s3-demo-bucket1.s3.us-west-2.amazonaws.com/puppy.png
```

Especificación de bucket de encabezado **`Host`** de HTTP

Siempre y cuando su solicitud `GET` no utilice el punto de enlace de SSL, puede especificar el bucket para la solicitud con el encabezado del `Host` HTTP. El encabezado del `Host` en una solicitud de REST se interpreta de la siguiente manera:

- Si se omite el encabezado `Host` o si su valor es `s3.`*`region-code`*`.amazonaws.com`, el bucket de la solicitud será el primer componente delimitado por una barra inclinada del URI de la solicitud y la clave para la solicitud será el resto del URI de la solicitud. Este es el método más común, según se ilustra en el primer y segundo ejemplo de esta sección. Solo se puede omitir el encabezado `Host` para solicitudes HTTP 1.0.
- De lo contrario, si el valor del encabezado `Host` termina en `.s3.`*`region-code`*`.amazonaws.com`, el nombre del bucket es la primera parte del valor del encabezado `Host` hasta `.s3.`*`region-code`*`.amazonaws.com`. La clave para la solicitud es el URI de la solicitud. Esta interpretación expone los buckets como subdominios de `.s3.`*`region-code`*`.amazonaws.com`, según se ilustra en los ejemplos tercero y cuarto de esta sección.
- De lo contrario, el bucket para la solicitud es el valor en minúscula del encabezado del `Host` y la clave para la solicitud es el URI de la solicitud. Esta interpretación es útil si registró el mismo nombre de DNS que el nombre de su bucket y estableció ese nombre como alias de nombre canónico (CNAME) para Amazon S3. El procedimiento para registrar nombres de dominio y configurar registros DNS de CNAME está fuera del propósito de esta guía, pero el resultado se muestra en el ejemplo final en esta sección.

Ejemplos

En esta sección se proporcionan ejemplos de URL y solicitudes.

Example : solicitudes y URL de tipo ruta

En este ejemplo se utiliza lo siguiente:

- Nombre del bucke - `example.com`
- Región: EE. UU. Este (Norte de Virginia)
- Nombre de clav - `homepage.html`

El URL es el siguiente:

```
http://s3.us-east-1.amazonaws.com/example.com/homepage.html
```

La solicitud es la siguiente:

```
GET /example.com/homepage.html HTTP/1.1
Host: s3.us-east-1.amazonaws.com
```

La solicitud con HTTP 1.0 y la omisión del encabezado de `Host` son los siguientes:

```
GET /example.com/homepage.html HTTP/1.0
```

Para obtener información acerca de los nombres compatibles con el DNS, consulte Limitaciones. Para obtener más información acerca de las claves, consulte Claves.

Example : solicitudes y URL de tipo host virtual

En este ejemplo se utiliza lo siguiente:

- Nombre del bucket: `amzn-s3-demo-bucket1`
- Región: Europa (Irlanda)
- Nombre de clave: `homepage.html`

El URL es el siguiente:

```
http://amzn-s3-demo-bucket1.s3.eu-west-1.amazonaws.com/homepage.html
```

La solicitud es la siguiente:

```
GET /homepage.html HTTP/1.1
Host: amzn-s3-demo-bucket1.s3.eu-west-1.amazonaws.com
```

Example : método de alias CNAME

Para utilizar este método, debe configurar su nombre de DNS como un alias de CNAME para *`bucket-name`*`.s3.us-east-1.amazonaws.com`. Para obtener más información, consulte Personalización de URL de Amazon S3 con registros CNAME.

En este ejemplo se utiliza lo siguiente:

- Nombre del bucke - `example.com`
- Nombre de clave: `homepage.html`

El URL es el siguiente:

```
http://www.example.com/homepage.html
```

El ejemplo es el siguiente:

```
GET /homepage.html HTTP/1.1
Host: www.example.com
```

Personalización de URL de Amazon S3 con registros CNAME

Según sus necesidades, es posible que no desee que `s3.`*`region-code`*`.amazonaws.com` aparezca en su sitio web o servicio. Por ejemplo, si aloja las imágenes del sitio web en Amazon S3, puede que prefiera `http://images.example.com/` en lugar de `http://images.example.com.s3.us-east-1.amazonaws.com/`. Cualquier bucket con un nombre compatible con el DNS se puede denominar de la siguiente manera: `http://`*`BucketName`*`.s3.`*`Region`*`.amazonaws.com/[`*`Filename`*`]`, por ejemplo, `http://images.example.com.s3.us-east-1.amazonaws.com/mydog.jpg`. Al utilizar CNAME, puede asignar `images.example.com` a un nombre de host de Amazon S3 para que la URL anterior pueda convertirse en `http://images.example.com/mydog.jpg`.

El nombre del bucket debe ser el mismo que el CNAME. Por ejemplo, si crea un CNAME para asignar `images.example.com` a `images.example.com.s3.us-east-1.amazonaws.com`, `http://images.example.com/filename` y `http://images.example.com.s3.us-east-1.amazonaws.com/filename` serán iguales.

El registro de DNS de CNAME debe usar el nombre de host de tipo alojamiento virtual adecuado como el alias de su nombre de dominio. Por ejemplo, si el nombre del bucket y el nombre de dominio son `images.example.com` y el bucket está en la región EE. UU. Este (Norte de Virginia), el registro CNAME debe tomar como 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 utiliza el nombre de host para determinar el nombre del bucket. Por tanto, el nombre del bucket debe ser el mismo que el CNAME. Por ejemplo, suponga que configuró `www.example.com`

como un CNAME para `www.example.com.s3.us-east-1.amazonaws.com`. Cuando accede a `http://www.example.com`, Amazon S3 recibe una solicitud similar a la siguiente:

Example

```
GET / HTTP/1.1
Host: www.example.com
Date: date
Authorization: signatureValue
```

Amazon S3 solo ve el nombre de host original `www.example.com` y desconoce el mapeo de CNAME que se utiliza para resolver la solicitud.

Se puede utilizar cualquier punto de conexión de Amazon S3 en un alias CNAME. Por ejemplo, se puede utilizar `s3.ap-southeast-1.amazonaws.com` en los alias CNAME. Para obtener más información acerca de los puntos de enlace, consulte Puntos de enlace de solicitud. Para crear un sitio web estático mediante un dominio personalizado, consulte Tutorial: Configuración de un sitio web estático mediante un dominio personalizado registrado con Route 53.

Important

Cuando utilice URL personalizadas con CNAME, tendrá que asegurarse de que existe un bucket coincidente para cualquier registro CNAME o alias que configure. Por ejemplo, si crea entradas DNS para `www.example.com` y `login.example.com` para publicar contenido web mediante S3, tendrá que crear los buckets `www.example.com` y `login.example.com`.

Cuando se configura un CNAME o registros de alias que apuntan a un punto de conexión de S3 sin un bucket coincidente, cualquier usuario de AWS puede crear ese bucket y publicar contenido con el alias configurado, aunque la propiedad no sea la misma.

Por el mismo motivo, le recomendamos que cambie o elimine el CNAME o el alias correspondiente cuando elimine un bucket.

Cómo asociar un nombre de host a un bucket de Amazon S3

Para asociar un nombre de host a un bucket de Amazon S3 mediante un alias CNAME

1. Seleccione un nombre de host que pertenezca a un dominio que usted controle.

   En este ejemplo se utiliza el subdominio `images` del dominio `example.com`.

2. Cree un bucket que coincida con el nombre de host.

   En este ejemplo, los nombres de host y de bucket son `images.example.com`. El nombre del bucket debe coincidir exactamente con el nombre de host.

3. Cree un registro DNS de CNAME que defina el nombre de host como un alias para el bucket de Amazon S3.

   Por ejemplo:

   ```
   images.example.com CNAME images.example.com.s3.us-west-2.amazonaws.com
   ```

   > **Important**
   >
   > Por motivos de direccionamiento de solicitudes, el registro de CNAME se debe definir exactamente como se muestra en el ejemplo anterior. De lo contrario, puede parecer que funciona correctamente, pero al final puede provocar un comportamiento impredecible.

   El procedimiento para la configuración de los registros DNS de CNAME depende de su servidor de DNS o proveedor de DNS. Para obtener información específica, consulte la documentación de su servidor o contáctese con su proveedor.

Limitaciones

La compatibilidad con SOAP por HTTP está obsoleta, pero SOAP aún se encuentra disponible con HTTPS. Las características nuevas de Amazon S3 no son compatibles con SOAP. En vez de usar SOAP, le recomendamos que utilice la API de REST o los SDK de AWS.

Compatibilidad con versiones anteriores

Las siguientes secciones abarcan varios aspectos de la compatibilidad con versiones anteriores de Amazon S3 que se relacionan con las solicitudes de URL de tipo ruta y de host virtual.

Puntos de conexión heredados

Algunas regiones admiten puntos de enlace heredados. Es posible que vea estos puntos de enlace en los registros de acceso al servidor o en los registros de AWS CloudTrail. Para obtener más información, consulte la siguiente información. Para obtener una lista completa de las regiones y

los puntos de conexión de Amazon S3, consulte Puntos de conexión y cuotas de Amazon S3 en la Referencia general de Amazon Web Services.

> **Important**
>
> Aunque es posible que vea los puntos de enlace heredados en los registros, es recomendable que utilice siempre la sintaxis estándar de puntos de enlace para obtener acceso a los buckets.
> Las URL de tipo host virtual de Amazon S3 utilizan el siguiente formato:
>
> ```
> https://bucket-name.s3.region-code.amazonaws.com/key-name
> ```
>
> En Amazon S3, las URL de tipo ruta utilizan el siguiente formato:
>
> ```
> https://s3.region-code.amazonaws.com/bucket-name/key-name
> ```

Región de S3

Algunas regiones antiguas de Amazon S3 admiten puntos de conexión que contienen un guion (`-`) entre `s3` y el código de región (por ejemplo, `s3-us-west-2`), en lugar de un punto (por ejemplo, `s3.us-west-2`). Si el bucket se encuentra en una de estas regiones, es posible que vea el siguiente formato de punto de enlace en los registros de acceso al servidor o en los registros de CloudTrail:

```
https://bucket-name.s3-region-code.amazonaws.com
```

En este ejemplo, el nombre del bucket es amzn-s3-demo-bucket1 y la región es EE. UU. Oeste (Oregón):

```
https://amzn-s3-demo-bucket1.s3-us-west-2.amazonaws.com
```

Punto de conexión global heredado

En algunas regiones, puede utilizar el punto de conexión global heredado para crear solicitudes que no especifiquen un punto de conexión específico de la región. El punto de enlace global heredado es el siguiente:

```
bucket-name.s3.amazonaws.com
```

En los registros de acceso al servidor o en los registros de CloudTrail, es posible que vea solicitudes que utilizan el punto de enlace global heredado. En este ejemplo, el nombre del bucket es `amzn-s3-demo-bucket1` y el punto de conexión global heredado es:

```
https://amzn-s3-demo-bucket1.s3.amazonaws.com
```

Solicitudes de tipo host virtual para Este de EE. UU. (Norte de Virginia)

Las solicitudes realizadas con el punto de conexión global heredado se envían a la región Este de EE. UU. (Norte de Virginia) de forma predeterminada. Por lo tanto, el punto de enlace global heredado se utiliza a veces como sustituto del punto de enlace regional para EE. UU. Este (Norte de Virginia). Si crea un bucket en EE. UU. Este (Norte de Virginia) y utiliza el punto de enlace, Amazon S3 envía su solicitud a esta región de forma predeterminada.

Solicitudes de tipo host virtual para otras regiones

El punto de conexión global heredado también se utiliza para solicitudes de tipo host virtual en otras regiones admitidas. Si crea un bucket en una región que se lanzó antes del 20 de marzo de 2019 y utiliza el punto de conexión global heredado, Amazon S3 actualiza el registro DNS para redirigir la solicitud a la ubicación correcta, lo que podría tardar algún tiempo. Mientras tanto, se aplica la regla predeterminada y su solicitud de tipo alojamiento virtual se dirige a la región de Este de EE. UU. (Norte de Virginia). A continuación, Amazon S3 la redirecciona con el redireccionamiento temporal HTTP 307 a la región correcta.

Para los buckets de S3 de regiones lanzadas después del 20 de marzo de 2019, el servidor DNS no enruta la solicitud directamente a la Región de AWS en la que se encuentra el bucket. En su lugar, devuelve un error de solicitud errónea HTTP 400. Para obtener más información, consulte Realizar solicitudes.

Solicitudes de tipo ruta

Para la región Este de EE. UU. (Norte de Virginia), el punto de conexión global heredado se puede utilizar para solicitudes de tipo ruta.

Para todas las demás regiones, la sintaxis de tipo ruta requiere que se use el punto de enlace específico de la región al intentar obtener acceso al bucket. Si intenta obtener acceso a un bucket con el punto de conexión global heredado u otro punto de conexión diferente al de la región donde reside el bucket, recibirá un error de redireccionamiento temporal con el código de respuesta HTTP

307 y un mensaje que indica el URI correcto para el recurso. Por ejemplo, si utiliza `https://s3.amazonaws.com/bucket-name` para un bucket creado en la región EE. UU. Oeste (Oregón), recibirá un error de redireccionamiento temporal HTTP 307.

Solicitar redireccionamiento y la API de REST

Temas

En esta sección se describe cómo administrar el redireccionamiento HTTP mediante la API de REST de Amazon S3. Para obtener información general sobre los redireccionamientos de Amazon S3, consulte Realizar solicitudes en la Referencia de la API de Amazon Simple Storage Service.

Redireccionamientos y agentes de usuario HTTP

Los programas que usan la API de REST de Amazon S3 deberían poder ocuparse de los redireccionamientos o bien en la capa de aplicación o bien en la capa HTTP. Muchas bibliotecas clientes y agentes de usuario HTTP pueden configurarse para administrar los redireccionamientos de forma correcta y automática. Sin embargo, hay muchas otras que tienen implementaciones de los redireccionamientos incorrectas o incompletas.

Antes de depender de una biblioteca para cumplir el requisito de redireccionamiento, pruebe lo siguiente:

- Compruebe que todos los encabezados de solicitudes HTTP estén incluidos correctamente en la solicitud redirigida (la segunda solicitud tras haber recibido un redireccionamiento) incluidos los estándares HTTP como la autorización y la fecha.
- Compruebe que los redireccionamientos que no sean GET, como PUT o DELETE, funcionen correctamente.
- Compruebe que las solicitudes PUT grandes sigan los redireccionamientos correctamente.
- Compruebe que las solicitudes PUT sigan los redireccionamientos correctamente si la respuesta 100-continue tarda demasiado en llegar.

Los agentes de usuario HTTP que se ajusten estrictamente a RFC 2616 podrían necesitar una confirmación explícita antes de seguir un redireccionamiento si el método de solicitud HTTP no es

GET o HEAD. Normalmente, es seguro seguir los redireccionamientos generados por Amazon S3 automáticamente, ya que el sistema generará redireccionamientos solamente a los hosts del dominio amazonaws.com, y el efecto de la solicitud redirigida será el mismo que el de la solicitud original.

Redireccionamientos y 100-continue

Para administrar el redireccionamiento de forma más sencilla, mejorar la eficacia y evitar los costos asociados con el envío duplicado del cuerpo de una solicitud redirigida, configure su aplicación para usar 100-continues en operaciones PUT. Cuando su aplicación usa 100-continue, no envía el cuerpo de la solicitud hasta que recibe una confirmación. Si el mensaje se rechaza en función de los encabezados, el cuerpo del mismo no se llega a enviar. Para obtener más información sobre 100-continue, visite RFC 2616 Sección 8.2.3

> **Note**
>
> Según RFC 2616, al usar `Expect: Continue` sin un servidor HTTP conocido, no debería esperar un periodo indefinido para enviar el cuerpo de la solicitud. Esto se debe a que algunos servidores HTTP no reconocen 100-continue. Sin embargo, Amazon S3 sí que reconoce si su solicitud contiene un `Expect: Continue` y responderá con un estado 100-continue provisional o un código de estado final. Además, no se producirá ningún error de redireccionamiento tras recibir la autorización provisional del 100-continue. Así, le resultará más fácil evitar recibir una respuesta de redireccionamiento mientras sigue escribiendo el cuerpo de la solicitud.

Ejemplo de redireccionamiento

En esta sección se ofrece un ejemplo de interacción cliente-servidor mediante redireccionamientos HTTP y 100-continue.

A continuación mostramos un ejemplo de PUT al bucket `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 devuelve lo siguiente:

```
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>
```

El cliente sigue la respuesta de redireccionamiento y envía una nueva solicitud al punto de conexión temporal `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 devuelve un 100-continue que indica que el cliente debería continuar con el envío del cuerpo de la solicitud.

```
HTTP/1.1 100 Continue
```

El cliente envía el cuerpo de la solicitud.

```
ha ha\n
```

Amazon S3 devuelve la respuesta final.

```
HTTP/1.1 200 OK
Date: Mon, 15 Oct 2007 22:18:48 GMT

ETag: "a2c8d6b872054293afd41061e93bc289"
Content-Length: 0
Server: AmazonS3
```

Desarrollo con Amazon S3 mediante la AWS CLI

Siga estos pasos para descargar y configurar AWS Command Line Interface (AWS CLI).

Para obtener una lista de los comandos de la AWS CLI de Amazon S3, consulte las siguientes páginas en la Referencia de comandos de la AWS CLI:

- s3
- s3api
- s3control

> **Note**
>
> Para poder acceder a los servicios de AWS, como Amazon S3, debe proporcionar credenciales. A continuación, el servicio puede determinar si usted tiene permisos para obtener acceso a sus recursos. La consola requiere que especifique la contraseña. Puede crear claves de acceso para su Cuenta de AWS con el fin de tener acceso a la AWS CLI o a la API. Sin embargo, no es recomendable acceder a AWS con las credenciales de su Cuenta de AWS. En su lugar, le recomendamos que utilice AWS Identity and Access Management (IAM). Cree un usuario de IAM, añada el usuario a un grupo de IAM con permisos administrativos y, a continuación, conceda permisos administrativos al usuario de IAM que ha creado. De este modo, podrá tener acceso a AWS mediante una URL especial y las credenciales de ese usuario de IAM. Para obtener instrucciones, consulte Creación del primer grupo de usuarios y administradores de IAM en la Guía del usuario de IAM.

Para configurar la AWS CLI

1. Descargue y configure la AWS CLI. Para obtener instrucciones, consulte los siguientes temas en la Guía del usuario de AWS Command Line Interface:

- Configuración inicial de la AWS Command Line Interface
- Configuración del AWS Command Line Interface

2. Añada un perfil con nombre para el usuario administrador en el archivo de configuración de la AWS CLI. Puede utilizar este perfil cuando ejecute los comandos de la AWS CLI. Para obtener más información, consulte Perfiles con nombre para la AWS CLI en la Guía del usuario de AWS Command Line Interface.

```
[adminuser]
aws_access_key_id = adminuser access key ID
aws_secret_access_key = adminuser secret access key
region = aws-region
```

Para ver una lista de las Regiones de AWS disponibles, consulte Regiones y puntos de conexión en la Referencia general de AWS.

3. Verifique la configuración escribiendo los siguientes comandos en el símbolo del sistema.

- Pruebe el comando `help` para verificar que la AWS CLI está instalada en su equipo:

```
aws help
```

- Ejecute un comando `S3` con las credenciales de `adminuser` que acaba de crear. Para ello, añada el parámetro `--profile` al comando para especificar el nombre del perfil. En este ejemplo, el comando `ls` muestra los buckets de su cuenta. La AWS CLI utiliza las credenciales de `adminuser` para autenticar la solicitud.

```
aws s3 ls --profile adminuser
```

Desarrollo con Amazon S3 mediante los SDK de AWS

Los kits de desarrollo de software (SDK) de AWS se encuentran disponibles en muchos lenguajes de programación populares. Cada SDK proporciona una API, ejemplos de código y documentación que facilitan a los desarrolladores la creación de aplicaciones en su lenguaje preferido.

> **Note**
>
> Puede utilizar AWS Amplify para el desarrollo integral y de pila completa de aplicaciones web y móviles. Amplify Storage integra a la perfección las capacidades de almacenamiento y administración de archivos en aplicaciones frontend web y móviles, basadas en Amazon S3. Para obtener más información, consulte Almacenamiento en la Guía del usuario de Amplify.

Uso de este servicio con un SDK de AWS

Los kits de desarrollo de software (SDK) de AWS se encuentran disponibles en muchos lenguajes de programación populares. Cada SDK proporciona una API, ejemplos de código y documentación que facilitan a los desarrolladores la creación de aplicaciones en su lenguaje preferido.

Documentación de SDK	Ejemplos de código
AWS SDK for C++	Ejemplos de código de AWS SDK for C++
AWS CLI	Ejemplos de código de AWS CLI
AWS SDK for Go	Ejemplos de código de AWS SDK for Go
AWS SDK for Java	Ejemplos de código de AWS SDK for Java
AWS SDK for JavaScript	Ejemplos de código de AWS SDK for JavaScript
AWS SDK para Kotlin	Ejemplos de código de AWS SDK para Kotlin
AWS SDK for .NET	Ejemplos de código de AWS SDK for .NET
AWS SDK for PHP	Ejemplos de código de AWS SDK for PHP
AWS Tools for PowerShell	Ejemplos de código de Herramientas para PowerShell
AWS SDK for Python (Boto3)	Ejemplos de código de AWS SDK for Python (Boto3)
AWS SDK for Ruby	Ejemplos de código de AWS SDK for Ruby

Documentación de SDK	Ejemplos de código
AWS SDK para Rust	Ejemplos de código de AWS SDK para Rust
AWS SDK para SAP ABAP	Ejemplos de código de AWS SDK para SAP ABAP
AWS SDK para Swift	Ejemplos de código de AWS SDK para Swift

Para obtener ejemplos específicos de este servicio, consulte Ejemplos de código de Amazon S3 con SDK de AWS.

> **Ejemplo de disponibilidad**
>
> ¿No encuentra lo que necesita? Solicite un ejemplo de código a través del enlace de Enviar comentarios que se encuentra al final de esta página.

Interfaces de programación del SDK

Cada SDK de AWS proporciona una o varias interfaces de programación para trabajar con Amazon S3. Cada SDK ofrece una interfaz de bajo nivel para Amazon S3, con métodos que imitan del modo más parecido posible las operaciones de la API. Algunos SDK ofrecen interfaces de alto nivel para Amazon S3, que son abstracciones diseñadas para simplificar los casos de uso frecuentes.

Por ejemplo, cuando realiza una carga de varias partes mediante las operaciones de la API de bajo nivel, debe utilizar una operación para iniciar la carga, otra operación para cargar las partes y una operación final para completar la carga. Una operación de la API de carga de varias partes de alto nivel le permite realizar todas las operaciones necesarias para la carga en una sola llamada a la API. Para ver ejemplos, consulte Carga de un objeto con la carga multiparte.

Las operaciones de la API de bajo nivel permiten un mayor control sobre la carga. Le recomendamos que use las operaciones de la API de bajo nivel si necesita detener y reanudar las cargas, variar los tamaños de las partes durante la carga o iniciar las cargas si no conoce de antemano el tamaño de los datos de carga.

Especificación de Signature Version en la autenticación de solicitudes

Amazon S3 solo es compatible con AWS Signature Version 4 en la mayoría de las Regiones de AWS. En algunas de las Regiones de AWS más antiguas, Amazon S3 admite Signature Version 4 y Signature Version 2. Sin embargo, Signature Version 2 se va a desactivar (esta característica quedará obsoleta). Para obtener más información sobre el final del periodo de soporte de Signature Version 2, consulte AWS Signature Version 2 se va a desactivar (quedará obsoleta) para Amazon S3.

Para ver una lista de todas las regiones de Amazon S3 y las versiones de Signature que admiten, consulte Regiones y puntos de enlace en la Referencia general de AWS.

Para todas las Regiones de AWS, los SDK de AWS utilizan Signature Version 4 de forma predeterminada con el fin de autenticar solicitudes. Si utiliza los SDK de AWS lanzados antes de mayo de 2016, es posible que deba solicitar Signature Version 4, como se muestra en la siguiente tabla.

SDK	Solicitud de Signature Version 4 para una autenticación de solicitud
AWS CLI	Para el perfil predeterminado, ejecute el siguiente comando: `$ aws configure set default.s3.signature_version s3v4` Para un perfil común, ejecute el siguiente comando: `$ aws configure set profile.your_profile_name.s3.signature_version s3v4`
SDK de Java	Añada lo siguiente en su código: `System.setProperty(SDKGlobalConfiguration.ENABLE_S3_SIGV4_SYSTEM_PROPERTY, "true");` O bien, en la línea de comando, especifique lo siguiente: `-Dcom.amazonaws.services.s3.enableV4`

SDK	Solicitud de Signature Version 4 para una autenticación de solicitud
SDK de JavaScript	Configure el parámetro `signatureVersion` en `v4` cuando cree el cliente: `var s3 = new AWS.S3({signatureVersion: 'v4'});`
SDK de PHP	Configure el parámetro `signature` en `v4` cuando cree el cliente del servicio de Amazon S3 para PHP SDK v2: `<?php` `$client = S3Client::factory([` `'region' => 'YOUR-REGION',` `'version' => 'latest',` `'signature' => 'v4'` `]);` Cuando utilice el SDK de PHP v3, establezca el parámetro `signature_version` en `v4` durante la construcción del cliente de servicio de Amazon S3: `<?php` `$s3 = new Aws\S3\S3Client([` `'version' => '2006-03-01',` `'region' => 'YOUR-REGION',` `'signature_version' => 'v4'` `]);`
SDK de Python-Boto	Especifique lo siguiente en el archivo de configuración predeterminado boto: `[s3] use-sigv4 = True`

SDK	Solicitud de Signature Version 4 para una autenticación de solicitud
SDK de Ruby	SDK de Ruby - Version 1: configure el parámetro `:s3_signature_version` en `:v4` cuando cree el cliente: `s3 = AWS::S3::Client.new(:s3_signature_version => :v4)` SDK de Ruby - Version 3: configure el parámetro `signature_version` en `v4` cuando cree el cliente: `s3 = Aws::S3::Client.new(signature_version: 'v4')`
SDK de .NET	Añada lo siguiente al código antes de crear el cliente de Amazon S3: `AWSConfigsS3.UseSignatureVersion4 = true;` O bien, añada lo siguiente al archivo de configuración: `<appSettings> <add key="AWS.S3.UseSignatureVersion4" value="true" /> </appSettings>`

AWS Signature Version 2 se va a desactivar (quedará obsoleta) para Amazon S3

Sin embargo, Signature Version 2 se va a desactivar (quedará obsoleta) en Amazon S3. Amazon S3 solo aceptará solicitudes de API que estén firmadas con Signature Version 4.

En esta sección, se incluyen algunas respuestas a preguntas comunes sobre el final del servicio de soporte de Signature Version 2.

¿Qué es Signature Version 2/4 y qué significa la firma de solicitudes?

El proceso de firma Signature Version 2 o Signature Version 4 se utiliza para autenticar las solicitudes de API de Amazon S3. La firma de solicitudes permite que Amazon S3 pueda identificar quién está enviando la solicitud y ayuda a proteger las solicitudes frente a agentes malintencionados.

Para obtener más información sobre la firma de solicitudes de AWS, consulte Firma de solicitudes de AWS en la Referencia general de AWS.

¿En qué consiste la actualización?

En la actualidad, pueden utilizarse solicitudes de la API de Amazon S3 firmadas con Signature Version 2 y Signature Version 4. Cuando esto suceda, Amazon S3 solamente aceptará solicitudes que estén firmadas con Signature Version 4.

Para obtener más información sobre la firma de solicitudes de AWS, consulte Cambios de Signature Version 4 en la Referencia general de AWS.

¿Por qué se realiza la actualización?

En lugar de utilizar una clave de acceso secreta, Signature Version 4 usa una clave de firma, lo que mejora la seguridad. En la actualidad, Signature Version 4 puede utilizarse en todas las Regiones de AWS, mientras que Signature Version 2 solo se admite en las regiones que se lanzaron antes de enero de 2014. Esta actualización nos permite proporcionar una experiencia más uniforme en todas las regiones.

¿Cómo sé si estoy utilizando Signature Version 4 y qué actualizaciones tengo que hacer?

Normalmente, la versión de Signature que se utiliza para firmar las solicitudes viene determinada por la herramienta o el SDK del lado del cliente. De forma predeterminada, las últimas versiones de los SDK de AWS utilizan Signature Version 4. En el caso del software de terceros, póngase en contacto con el equipo de soporte del software correspondiente para confirmar la versión que necesita. Si envía llamadas REST directas a Amazon S3, debe modificar la aplicación para que utilice el proceso de firma Signature Version 4.

Para obtener información acerca de qué versión de los SDK de AWS debe usarse al pasar a Signature Version 4, consulte Transición de Signature Version 2 a Signature Version 4.

Para obtener información acerca del uso de Signature Version 4 con la API de REST de Amazon S3, consulte Autenticación de solicitudes (AWS Signature Version 4) en la Referencia de la API de Amazon Simple Storage Service.

¿Qué sucede si no realizo la actualización?

Las solicitudes firmadas con Signature Version 2 que se generen cuando esto suceda, no podrán autenticarse con Amazon S3. Los solicitantes recibirán mensajes de error en los que se informará de que la solicitud debe firmarse con Signature Version 4.

¿Debo hacer algún cambio aunque utilice una URL prefirmada que requiera mi firma durante más de siete días?

Si utiliza una URL prefirmada que necesita su firma durante más de siete días, no tiene que hacer nada por el momento. Podrá seguir utilizando AWS Signature Version 2 para firmar y autenticar la URL prefirmada. Seguiremos investigando y le proporcionaremos más detalles sobre la migración a Signature Version 4 con direcciones URL prefirmadas.

Más información

- Para obtener más información acerca del uso de Signature Version 4, consulte Firma de solicitudes de la API de AWS.
- Consulte la lista de cambios entre Signature Version 2 y Signature Version 4 en Cambios de Signature Version 4.
- Consulte la publicación AWS Signature Version 4 to replace AWS Signature Version 2 for signing Amazon S3 API requests en los foros de AWS.
- Si tiene alguna pregunta o duda, contáctese con AWS Support.

Transición de Signature Version 2 a Signature Version 4

Si en la actualidad utiliza Signature Version 2 para autenticar las solicitudes de API de Amazon S3, debe cambiar a Signature Version 4. Tal y como se describe en , el servicio de soporte de Signature Version 2 está a punto de finaliza AWS Signature Version 2 se va a desactivar (quedará obsoleta) para Amazon S3.

Para obtener información acerca del uso de Signature Version 4 con la API de REST de Amazon S3, consulte Autenticación de solicitudes (AWS Signature Version 4) en la Referencia de la API de Amazon Simple Storage Service.

En la tabla siguiente, se muestran los SDK con la versión mínima necesaria para utilizar Signature Version 4 (SigV4). Si utiliza URL prefirmadas con los SDK de AWS para Java, JavaScript (Node.js) o Python (Boto/CLI), debe especificar la Región de AWS apropiada y configurar Signature Version 4

para el cliente. Para obtener información acerca de la configuración de `SigV4` en el cliente, consulte Especificación de Signature Version en la autenticación de solicitudes.

Si utiliza este SDK o producto	Actualice a esta versión del SDK	¿Es necesario cambiar el código para que el cliente utilice Sigv4?	Enlace a la documentación del SDK
AWS SDK for Java v1	Actualice a Java 1.11.201+ o v2.	Sí	Especificación de Signature Version en la autenticación de solicitudes
AWS SDK for Java v2	No es necesario actualizar el SDK.	No	AWS SDK for Java
AWS SDK for .NET v1	Actualice a 3.1.10 o versiones posteriores.	Sí	AWS SDK for .NET
AWS SDK for .NET v2	Actualice a 3.1.10 o versiones posteriores.	No	AWS SDK for .NET v2
AWS SDK for .NET v3	Actualice a 3.3.0.0 o versiones posteriores.	Sí	AWS SDK for .NET v3
AWS SDK for JavaScript v1	Actualice a 2.68.0 o	Sí	AWS SDK for JavaScript

Si utiliza este SDK o producto	Actualice a esta versión del SDK	¿Es necesario cambiar el código para que el cliente utilice Sigv4?	Enlace a la documentación del SDK
	versiones posteriores.		
AWS SDK for JavaScript v2	Actualice a 2.68.0 o versiones posteriores.	Sí	AWS SDK for JavaScript
AWS SDK for JavaScript v3	No tiene que hacer nada por el momento. Actualice a la versión principal V3 en el tercer trimestre de 2019.	No	AWS SDK for JavaScript

Si utiliza este SDK o producto	Actualice a esta versión del SDK	¿Es necesario cambiar el código para que el cliente utilice Sigv4?	Enlace a la documentación del SDK
AWS SDK for PHP v1	Se recomienda actualizar a la versión más reciente de PHP o, al menos, a la v2.7.4 con el parámetro de firma establecido en v4 en la configuración del cliente de S3.	Sí	AWS SDK for PHP
AWS SDK for PHP v2	Se recomienda actualizar a la versión más reciente de PHP o, al menos, a la v2.7.4 con el parámetro de firma establecido en v4 en la configuración del cliente de S3.	No	AWS SDK for PHP

Si utiliza este SDK o producto	Actualice a esta versión del SDK	¿Es necesario cambiar el código para que el cliente utilice Sigv4?	Enlace a la documentación del SDK
AWS SDK for PHP v3	No es necesario actualizar el SDK.	No	AWS SDK for PHP
Boto2	Actualice a Boto2 v2.49.0.	Sí	Actualización a Boto 2
Boto3	Actualice a 1.5.71 (Botocore), 1.4.6 (Boto3).	Sí	Boto 3: AWS SDK para Python
AWS CLI	Actualice a 1.11.108.	Sí	AWS Command Line Interface
AWS CLI v2 (vista previa)	No es necesario actualizar el SDK.	No	Versión 2 de la AWS Command Line Interface
AWS SDK for Ruby v1	Actualice a Ruby V3.	Sí	Ruby V3 para AWS
AWS SDK for Ruby v2	Actualice a Ruby V3.	Sí	Ruby V3 para AWS
AWS SDK for Ruby v3	No es necesario actualizar el SDK.	No	Ruby V3 para AWS

Si utiliza este SDK o producto	Actualice a esta versión del SDK	¿Es necesario cambiar el código para que el cliente utilice Sigv4?	Enlace a la documentación del SDK
Go	No es necesario actualizar el SDK.	No	AWS SDK for Go
C++	No es necesario actualizar el SDK.	No	AWS SDK for C++

AWS Tools for Windows PowerShell o bien AWS Tools for PowerShell Core

Si utiliza versiones del módulo anteriores a 3.3.0.0, debe actualizar a 3.3.0.0.

Para obtener información sobre la versión, utilice el cmdlet `Get-Module`:

```
Get-Module -Name AWSPowershell
Get-Module -Name AWSPowershell.NetCore
```

Para actualizar a la versión 3.3.0.0, utilice el cmdlet `Update-Module`:

```
Update-Module -Name AWSPowershell
Update-Module -Name AWSPowershell.NetCore
```

Puede utilizar direcciones URL prefirmadas que sean válidas durante más siete días en las que el tráfico se envíe con Signature Version 2.

Desarrollo con Amazon S3 mediante la API REST

La arquitectura de Amazon S3 está diseñada con un lenguaje de programación neutro y utiliza nuestras interfaces para almacenar y recuperar objetos.

Amazon S3 ofrece actualmente una interfaz REST. Con REST, los metadatos se devuelven en encabezados HTTP. Dado que solo admitimos solicitudes HTTP de hasta 4 KB (sin incluir el cuerpo), la cantidad de metadatos que puede aplicar es limitada. La API REST es una interfaz HTTP para Amazon S3. Con REST, usted puede utilizar solicitudes HTTP estándar para crear, recuperar y eliminar buckets y objetos.

Puede utilizar cualquier conjunto de herramientas que admita HTTP para utilizar la API REST. Incluso puede utilizar un navegador para recuperar objetos, siempre y cuando se puedan leer de forma anónima.

La API REST utiliza códigos de estado y encabezados HTTP estándar, para que los conjuntos de herramientas y los navegadores estándar funcionen según lo previsto. En algunas áreas, hemos añadido una funcionalidad al HTTP (por ejemplo, añadimos encabezados para admitir el control de acceso). En estos casos, hicimos todo lo posible para añadir la nueva funcionalidad de manera que coincida con el estilo del uso de HTTP estándar.

Para obtener más información sobre el envío de solicitudes mediante la API REST, consulte Realizar solicitudes con la API REST. Para obtener algunas consideraciones que debe tener en cuenta al utilizar la API REST, consulte los siguientes temas.

Para obtener más información sobre el uso de la API de REST de Amazon S3, consulte la referencia de la API de Amazon Simple Storage Service.

Temas

Enrutamiento de solicitudes

Los programas que realizan solicitudes a buckets creados con la API CreateBucket que incluyen una CreateBucketConfiguration deben admitir el redireccionamiento. Además, algunos clientes que no respetan los TTL de DNS podrían producir problemas.

En esta sección se describe el direccionamiento y algunas cuestiones sobre DNS que se deben tener en cuenta al usar Amazon S3.

Solicitar redireccionamiento y la API REST

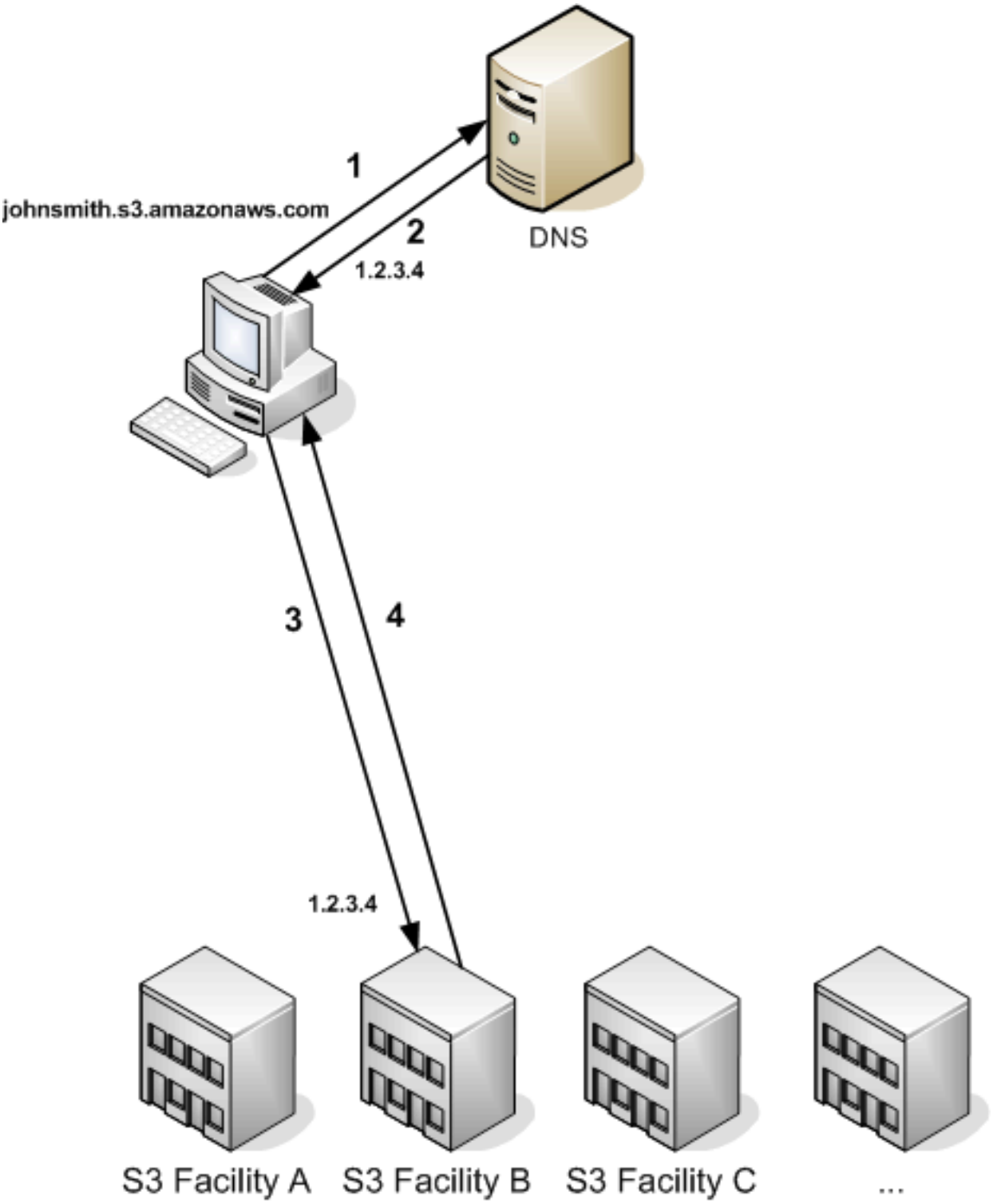
Amazon S3 utiliza el sistema de nombres de dominio (DNS) para dirigir las solicitudes a las ubicaciones que pueden procesarlas. Este sistema es eficaz, pero se pueden producir errores de direccionamiento temporal. Si una solicitud llega a la ubicación incorrecta de Amazon S3, Amazon S3 responde con un redireccionamiento temporal que le indica al solicitante que debe volver a enviar la solicitud a un nuevo punto de enlace. Si una solicitud se realiza incorrectamente, Amazon S3 utiliza el redireccionamiento permanente para proporcionar instrucciones sobre cómo realizar la solicitud correctamente.

> **Important**
>
> Para utilizar esta característica, debe tener una aplicación que pueda gestionar las respuestas de redirección de Amazon S3. La única excepción es para las aplicaciones que funcionan exclusivamente con buckets creados sin `<CreateBucketConfiguration>`. Para obtener más información acerca de las restricciones de ubicación, consulte Acceso y publicación de un bucket de Amazon S3.
>
> Para todas las regiones que se lanzaron después del 20 de marzo de 2019, si una solicitud llega a la ubicación de Amazon S3 incorrecta, Amazon S3 devuelve un error de solicitud errónea HTTP 400.
>
> Para obtener más información sobre cómo habilitar o deshabilitar una Región de AWS, consulte Puntos de conexión y Regiones de AWS en la Referencia general de AWS.

Temas

Enrutamiento de DNS

El direccionamiento del sistema de nombres de dominio (DNS) dirige las solicitudes a las ubicaciones correctas de Amazon S3. En la figura y el procedimiento siguientes se muestra un ejemplo de direccionamiento de DNS.



Pasos de la solicitud de direccionamiento de DNS

1. El cliente realiza una solicitud de DNS para obtener un objeto almacenado en Amazon S3.

2. El cliente recibe una o más direcciones IP para las ubicaciones que pueden procesar la solicitud. En este ejemplo, la dirección IP es para la ubicación B.
3. El cliente realiza una solicitud a la ubicación B de Amazon S3.
4. La ubicación B devuelve una copia del objeto al cliente.

Redireccionamiento temporal de solicitud

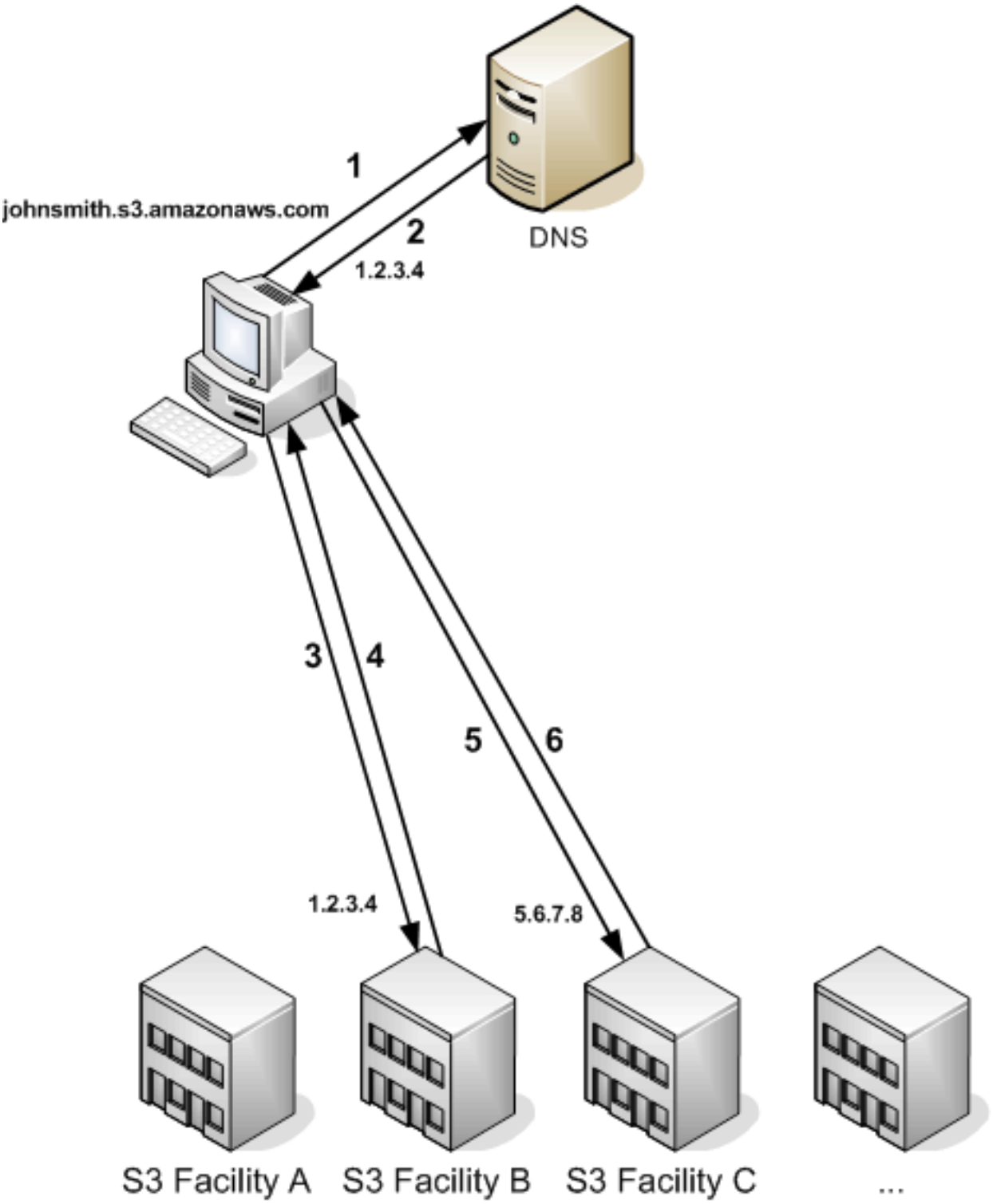
Un redireccionamiento temporal es un tipo de respuesta de error que indica al solicitante que debe volver a enviar la solicitud a otro punto de enlace. Debido a las características de distribución de Amazon S3, las solicitudes pueden dirigirse temporalmente a la ubicación incorrecta. Es posible que esto ocurra inmediatamente después de crear o eliminar buckets.

Por ejemplo, si crea un bucket nuevo y realiza una solicitud al bucket de inmediato, es posible que reciba un redireccionamiento temporal, según la restricción de ubicación del bucket. Si creó el bucket en la Región de AWS EE. UU. Este (Norte de Virginia), no verá el redireccionamiento, ya que este también es el punto de enlace predeterminado de Amazon S3.

Sin embargo, si el bucket se creó en otra región, cualquier solicitud para el bucket se dirigirá al punto de enlace predeterminado mientras la entrada de DNS del bucket se propaga. El punto de enlace predeterminado redirige la solicitud al punto de enlace correcto con una respuesta HTTP 302. El redireccionamiento temporal contiene un Uniform Resource Identifier (URI, Identificador de recursos uniforme) en la ubicación correcta, que usted puede utilizar para volver a enviar la solicitud de inmediato.

> **Important**
>
> No vuelva a utilizar un punto de enlace proporcionado por una respuesta de redireccionamiento anterior. Puede parecer que funciona (incluso durante largos períodos de tiempo), pero podría proporcionar resultados impredecibles y fallar sin previo aviso.

En la figura y el procedimiento siguientes se muestra un ejemplo de un redireccionamiento temporal.



Pasos del redireccionamiento de solicitud temporal

1. El cliente realiza una solicitud de DNS para obtener un objeto almacenado en Amazon S3.
2. El cliente recibe una o más direcciones IP para las ubicaciones que pueden procesar la solicitud.

3. El cliente realiza una solicitud a la ubicación B de Amazon S3.
4. La ubicación B devuelve un redireccionamiento que indica que el objeto está disponible en la ubicación C.
5. El cliente vuelve a enviar la solicitud a la ubicación C.
6. La ubicación C devuelve una copia del objeto.

Redireccionamiento permanente de solicitud

Un redireccionamiento permanente indica que su solicitud se dirigió a un recurso de forma incorrecta. Por ejemplo, el redireccionamiento permanente se produce si utiliza una solicitud de tipo ruta para obtener acceso a un bucket creado con `<CreateBucketConfiguration>`. Para obtener más información, consulte Acceso y publicación de un bucket de Amazon S3.

Para ayudar a detectar estos errores durante el desarrollo, este tipo de redireccionamiento no contiene un encabezado HTTP de ubicación que le permita seguir automáticamente la solicitud en la ubicación correcta. Consulte el documento de errores XML resultante para obtener ayuda sobre cómo usar el punto de enlace correcto de Amazon S3.

Ejemplos de redireccionamiento de solicitud

A continuación se proporcionan ejemplos de respuestas de redireccionamiento de solicitud temporal.

Respuesta de redireccionamiento temporal de la 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>
```

Respuesta de redireccionamiento temporal de la API de SOAP

> **Note**
>
> La compatibilidad con SOAP por HTTP está obsoleta, pero SOAP aún se encuentra disponible con HTTPS. Las características nuevas de Amazon S3 no son compatibles con SOAP. En vez de usar SOAP, le recomendamos que utilice la API de REST o los SDK de 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>
```

Consideraciones sobre DNS

Uno de los requisitos de diseño de Amazon S3 es una disponibilidad extremadamente alta. Una de las maneras de cumplir con este requisito es mediante la actualización de las direcciones IP asociadas con el punto de enlace de Amazon S3 en el DNS según sea necesario. Estos cambios se reflejan automáticamente en los clientes cuya vida útil es corta, pero no en algunos clientes cuya vida útil es larga. Aquellos clientes cuya vida útil es larga deben realizar acciones especiales para volver a resolver el punto de enlace de Amazon S3 periódicamente a fin de beneficiarse de estos cambios. Para obtener más información acerca de las máquinas virtuales (VM), consulte lo siguiente:

- Para Java, la Java Virtual Machine (JVM, Máquina virtual de Java) de Sun almacena en caché las búsquedas del DNS para siempre de forma predeterminada; consulte la sección de “InetAddress Caching” de la documentación de InetAddress para obtener información acerca de cómo cambiar este comportamiento.
- Para PHP, la VM persistente de PHP que ejecuta las más populares configuraciones de implementación almacena en caché las búsquedas del DNS hasta que la VM se reinicia. Consulte los documentos de PHP de getHostByName.

Gestionar errores de REST y SOAP

Temas

En esta sección se describen los errores de REST y SOAP, y cómo controlarlos.

> **Note**
>
> La compatibilidad con SOAP por HTTP está obsoleta, pero SOAP aún se encuentra disponible con HTTPS. Las características nuevas de Amazon S3 no son compatibles con SOAP. En vez de usar SOAP, le recomendamos que utilice la API de REST o los SDK de AWS.

La respuesta de error de REST

Si una solicitud REST genera un error, la respuesta HTTP incluye lo siguiente:

- Un documento de error XML como cuerpo de la respuesta.
- Tipo de contenido: aplicación/xml.
- Un código de estado HTTP apropiado de 3xx, 4xx o 5xx.

A continuación se muestra un ejemplo de una respuesta de error de 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>
```

Para obtener más información acerca de los errores de Amazon S3, consulte ErrorCodeList.

Encabezados de respuesta

Los siguientes son encabezados de respuesta que devuelven todas las operaciones:

- `x-amz-request-id:` Un ID único que el sistema asigna a cada solicitud. En el caso poco probable de que tenga problemas con Amazon S3, Amazon puede utilizar esto para ayudar a solucionar el problema.
- `x-amz-id-2:` Un token especial que nos ayudará a solucionar los problemas.

Respuesta de error

Cuando se genera un error en una solicitud de Amazon S3, el cliente recibe una respuesta de error. El formato exacto de la respuesta de error es específico de la API: por ejemplo, la respuesta de error de REST difiere de la respuesta de error de SOAP. Sin embargo, todas las respuestas de error tienen elementos en común.

> **Note**
>
> La compatibilidad con SOAP por HTTP está obsoleta, pero SOAP aún se encuentra disponible con HTTPS. Las características nuevas de Amazon S3 no son compatibles con SOAP. En vez de usar SOAP, le recomendamos que utilice la API de REST o los SDK de AWS.

Código de error

El código de error es una cadena que identifica de forma exclusiva una condición de error. Está diseñado para que los programas que detectan y administran errores por tipo puedan leerlo y comprenderlo. Muchos códigos de error son comunes entre las API de SOAP y REST, pero algunos son específicos de la API. Por ejemplo, NoSuchKey es universal, pero UnexpectedContent se puede producir únicamente en respuesta a una solicitud REST no válida. En todos los casos, los códigos de falla de SOAP incluyen un prefijo, como se indica en la tabla de códigos de error, por lo que un error de NoSuchKey se devuelve efectivamente en SOAP como Client.NoSuchKey.

> **Note**
>
> La compatibilidad con SOAP por HTTP está obsoleta, pero SOAP aún se encuentra disponible con HTTPS. Las características nuevas de Amazon S3 no son compatibles con

SOAP. En vez de usar SOAP, le recomendamos que utilice la API de REST o los SDK de AWS.

Mensaje de error

El mensaje de error contiene una descripción genérica de la condición de error en inglés. Está destinado a un público humano. Los programas simples muestran el mensaje directamente al usuario final si se detecta una condición de error que no sabe cómo controlar o no le interesa hacerlo. Los programas sofisticados con un control de errores más exhaustivo y una internacionalización adecuada tienen más probabilidades de ignorar el mensaje de error.

Más detalles

Muchas respuestas de error contienen datos estructurados adicionales diseñados para ser leídos y comprendidos por un desarrollador que diagnostica errores de programación. Por ejemplo, si envía un encabezado Content-MD5 con una solicitud PUT de REST que no coincide con el resumen calculado en el servidor, recibe el error BadDigest. La respuesta de error también incluye como elementos de detalle el resumen que calculamos y el resumen que usted nos anticipó. Durante el desarrollo, puede utilizar esta información para diagnosticar el error. En producción, un programa con buen comportamiento puede incluir esta información en el registro de error.

La respuesta de error de SOAP

> **Note**
>
> La compatibilidad con SOAP por HTTP está obsoleta, pero SOAP aún se encuentra disponible con HTTPS. Las características nuevas de Amazon S3 no son compatibles con SOAP. En vez de usar SOAP, le recomendamos que utilice la API de REST o los SDK de AWS.

En SOAP, el cliente recibe un resultado de error como una falla de SOAP, con el código de respuesta HTTP 500. Si no recibe una falla de SOAP, su solicitud se completó correctamente. El código de falla de SOAP de Amazon S3 consta de un código de falla SOAP 1.1 estándar (“Servidor” o “Cliente”) concatenado con el código de error específico de Amazon S3. Por ejemplo: “Server.InternalError” o “Client.NoSuchBucket”. El elemento de cadena de falla de SOAP incluye un mensaje de error genérico legible en inglés. Por último, el elemento de detalle de falla de SOAP incluye información variada relacionada con el error.

Por ejemplo, si intenta eliminar el objeto “Fred”, que no existe, el cuerpo de la respuesta de SOAP incluye una falla “NoSuchKey” de SOAP.

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>
```

Para obtener más información acerca de los errores de Amazon S3, consulte ErrorCodeList.

Prácticas recomendadas para los errores de Amazon S3

Cuando diseña una aplicación para utilizar con Amazon S3 es importante controlar correctamente los errores de Amazon S3. En esta sección se describen problemas que debe tener en cuenta al diseñar su aplicación.

Reintente en caso de recibir una respuesta de InternalErrors

Los errores internos son errores que se producen dentro del entorno de Amazon S3.

Es posible que las solicitudes que reciben una respuesta InternalError no se hayan procesado. Por ejemplo, si una solicitud PUT devuelve un error InternalError, una operación GET posterior puede recuperar el valor anterior o el valor actualizado.

Si Amazon S3 devuelve una respuesta de InternalError, repita la solicitud.

Ajustar la aplicación para errores SlowDown repetidos

Como en cualquier sistema distribuido, S3 tiene mecanismos de protección que detectan el consumo excesivo de recursos intencional o no intencional y reaccionan en consecuencia. Los errores de SlowDown se pueden producir cuando una velocidad de solicitud alta activa uno de estos mecanismos. La reducción de la velocidad de su solicitud disminuirá o eliminará errores de este tipo. En términos generales, la mayoría de los usuarios no experimentará estos errores de manera habitual. Sin embargo, si desea obtener más información o experimenta errores SlowDown repetidos

o imprevistos, publique en nuestro foro para desarrolladores de Amazon S3 o regístrese en AWS Support en https://aws.amazon.com/premiumsupport/.

Aislar los errores

> **Note**
>
> La compatibilidad con SOAP por HTTP está obsoleta, pero SOAP aún se encuentra disponible con HTTPS. Las características nuevas de Amazon S3 no son compatibles con SOAP. En vez de usar SOAP, le recomendamos que utilice la API de REST o los SDK de AWS.

Amazon S3 proporciona un conjunto de códigos de error que se utilizan en la API de SOAP y en la de REST. La API de SOAP devuelve códigos de error estándares de Amazon S3. La API de REST está diseñada para tener el aspecto de un servidor HTTP estándar e interactuar con clientes HTTP existentes (p. ej., navegadores, bibliotecas de clientes HTTP, servidores proxy, cachés, etc.). Para asegurarnos de que los clientes HTTP controlen los errores correctamente, a cada error de Amazon S3 le asignamos un código de estado HTTP.

Los códigos de estado HTTP son menos costosos que los códigos de error de Amazon S3 e incluyen menos información sobre el error. Por ejemplo, los errores `NoSuchKey` y `NoSuchBucket` de Amazon S3 corresponden al código de estado `HTTP 404 Not Found`.

Si bien los códigos de estado HTTP contienen menos información sobre el error, los clientes que comprenden el HTTP pero no la API de Amazon S3, por lo general, pueden controlar los errores correctamente.

Por lo tanto, al momento de controlar errores o informar errores de Amazon S3 a los usuarios finales, utilice el código de error de Amazon S3 en lugar del código de estado HTTP, ya que contiene más información sobre el error. Además, al depurar su aplicación, también debe consultar el elemento legible <Details> de la respuesta de error de XML.

Referencia para el desarrollador

Este apéndice incluye las siguientes secciones.

Temas

Apéndice A: Usar la API de SOAP

> **Note**
>
> La compatibilidad con SOAP por HTTP está obsoleta, pero SOAP aún se encuentra disponible con HTTPS. Las características nuevas de Amazon S3 no son compatibles con SOAP. En vez de usar SOAP, le recomendamos que utilice la API de REST o los SDK de AWS.

Esta sección contiene información específica sobre la API SOAP de Amazon S3.

> **Note**
>
> Las solicitudes SOAP, tanto autenticadas como anónimas, deben enviarse a Amazon S3 con SSL. Amazon S3 devuelve un error si envía una solicitud SOAP por HTTP.

Temas

Elementos comunes de la API de SOAP

> **Note**
>
> La compatibilidad con SOAP por HTTP está obsoleta, pero SOAP aún se encuentra disponible con HTTPS. Las características nuevas de Amazon S3 no son compatibles con SOAP. En vez de usar SOAP, le recomendamos que utilice la API de REST o los SDK de AWS.

Puede usar SOAP 1.1 en HTTP para interactuar con Amazon S3. El WSDL de Amazon S3, que describe la API de Amazon S3 en lenguaje máquina, está disponible en: https://

doc.s3.amazonaws.com/2006-03-01/AmazonS3.wsdl. El esquema de Amazon S3 está disponible en https://doc.s3.amazonaws.com/2006-03-01/AmazonS3.xsd.

La mayoría de los usuarios utilizan el conjunto de herramientas de SOAP adaptado para su lenguaje y entorno de desarrollo para interactuar con Amazon S3. Los diferentes conjuntos de herramientas exponen la API de Amazon S3 de diferentes maneras. Consulte la documentación de su conjunto de herramientas específico para comprender cómo usarlo. En esta sección se ilustran las operaciones de SOAP de Amazon S3 de forma independiente del conjunto de herramientas, para lo cual se exhiben las solicitudes y respuestas XML tal como aparecen "en la ruta".

Elementos comunes

Puede incluir los siguientes elementos relacionados a autorización con cualquier solicitud de SOAP:

- `AWSAccessKeyId:` el ID de clave de acceso de AWS del solicitante
- `Timestamp:` La hora actual en el sistema
- `Signature:` La firma de la solicitud

Autenticar solicitudes SOAP

> **Note**
>
> La compatibilidad con SOAP por HTTP está obsoleta, pero SOAP aún se encuentra disponible con HTTPS. Las características nuevas de Amazon S3 no son compatibles con SOAP. En vez de usar SOAP, le recomendamos que utilice la API de REST o los SDK de AWS.

Cada solicitud no anónima debe contener información de autenticación para establecer la identidad de la solicitud principal que hace la solicitud. En SOAP, la información de autenticación se coloca en los siguientes elementos de la solicitud SOAP:

- Su ID de clave de acceso de AWS

> **Note**
>
> Cuando se haces solicitudes autenticadas SOAP, no se admiten credenciales de seguridad temporales. Para obtener más información acerca de estos tipos de credenciales, consulte Realizar solicitudes.

- `Timestamp:` Debe ser una fecha y hora (consulte http://www.w3.org/TR/xmlschema-2/#dateTime) en la zona horaria del Tiempo universal coordinado (Tiempo medio de Greenwich), como `2009-01-01T12:00:00.000Z`. La autorización fallará si esta marca temporal está adelantada más de 15 minutos que el reloj de los servicios de Amazon S3.
- `Signature:` el resumen RFC 2104 HMAC-SHA1 (consulte http://www.ietf.org/rfc/rfc2104.txt) de la concatenación de "AmazonS3" + OPERATION + Timestamp, con su clave de acceso secreta deAWS como clave. Por ejemplo, en la siguiente solicitud de ejemplo CreateBucket, el elemento de firma debe contener el resumen HMAC-SHA1 del valor "AmazonS3CreateBucket2009-01-01T12:00:00.000Z":

Por ejemplo, en la siguiente solicitud de ejemplo CreateBucket, el elemento de firma debe contener el resumen HMAC-SHA1 del valor "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**
>
> Las solicitudes SOAP, tanto autenticadas como anónimas, deben enviarse a Amazon S3 con SSL. Amazon S3 devuelve un error si envía una solicitud SOAP por HTTP.

Important

Debido a diferentes interpretaciones con respecto a cómo se debe descartar la precisión de tiempo adicional, los usuarios de .NET deben tener cuidado y no enviar a Amazon S3 demasiadas marcas temporales específicas. Esto se puede lograr creando objetos `DateTime` manualmente con solo una precisión de milisegundos.

Configurar políticas de acceso con SOAP

Note

La compatibilidad con SOAP por HTTP está obsoleta, pero SOAP aún se encuentra disponible con HTTPS. Las características nuevas de Amazon S3 no son compatibles con SOAP. En vez de usar SOAP, le recomendamos que utilice la API de REST o los SDK de AWS.

El control de acceso se puede establecer en el momento en el que se escribe un bucket o un objeto incluyendo el elemento "AccessControlList" en la solicitud para `CreateBucket`, `PutObjectInline` o `PutObject`. El elemento "AccessControlList" se describe en Administración de identidades y accesos para Amazon S3. Si no se ha especificado ninguna lista de control de acceso con estas operaciones, el recurso se creará con una política de acceso predeterminada que otorga al solicitante un acceso FULL_CONTROL (este será el caso incluso aunque la solicitud sea del tipo PutObjectInline o PutObject para un objeto que ya exista).

A continuación presentamos una solicitud que escribe datos en un objeto, hace que el objeto sea legible desde principales anónimos y proporciona al usuario especificado derechos FULL_CONTROL sobre el bucket (la mayoría de desarrolladores querrán concederse a sí mismos un acceso FULL_CONTROL a su propio bucket).

Example

A continuación aparece una solicitud que escribe datos en un objeto y hace que el objeto sea legible desde principales anónimos.

`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>
```

Esta política de control de acceso se puede leer o configurar para un bucket o un objeto existentes mediante los métodos `GetBucketAccessControlPolicy`, `GetObjectAccessControlPolicy`, `SetBucketAccessControlPolicy` y `SetObjectAccessControlPolicy`. Para obtener más información, consulte la explicación detallada de estos métodos.

Apéndice B: autenticación de solicitudes (AWS Signature Version 2)

Important

En esa sección, se explica cómo se autentican solicitudes con AWS Signature Version 2. Signature Version 2 se va a desactivar (esta característica quedará obsoleta). Amazon S3 solo aceptará solicitudes de API que estén firmadas con Signature Version 4. Para obtener más información, consulte AWS Signature Version 2 se va a desactivar (quedará obsoleta) para Amazon S3

Signature Version 4 es compatible con todas las Regiones de AWS y es la única versión que puede utilizarse en las regiones nuevas. Para obtener más información, consulte Autenticación de solicitudes (AWS Signature Version 4) en la Referencia de la API de Amazon Simple Storage Service.

Amazon S3 le ofrece la posibilidad de identificar qué versión de la API de firma se utilizó para firmar una solicitud. Es importante que identifique si alguno de sus flujos de trabajo está utilizando la firma de Signature Version 2 y que los actualice para que utilicen Signature Version 4 con el fin de evitar que su negocio resulte afectado.

- Si utiliza los registros de eventos de CloudTrail (opción recomendada), consulte Identificación de solicitudes de firma de Amazon S3 versión 2 mediante CloudTrail para saber cómo buscar e identificar dichas solicitudes.
- Si utiliza los registros de acceso del servidor de Amazon S3, consulte Identificación de solicitudes de la versión 2 de firma mediante registros de acceso de Amazon S3

Temas

Autenticar solicitudes con la API REST

Cuando accede a Amazon S3 con REST, debe proporcionar los siguientes elementos en su solicitud para que esta se pueda autenticar:

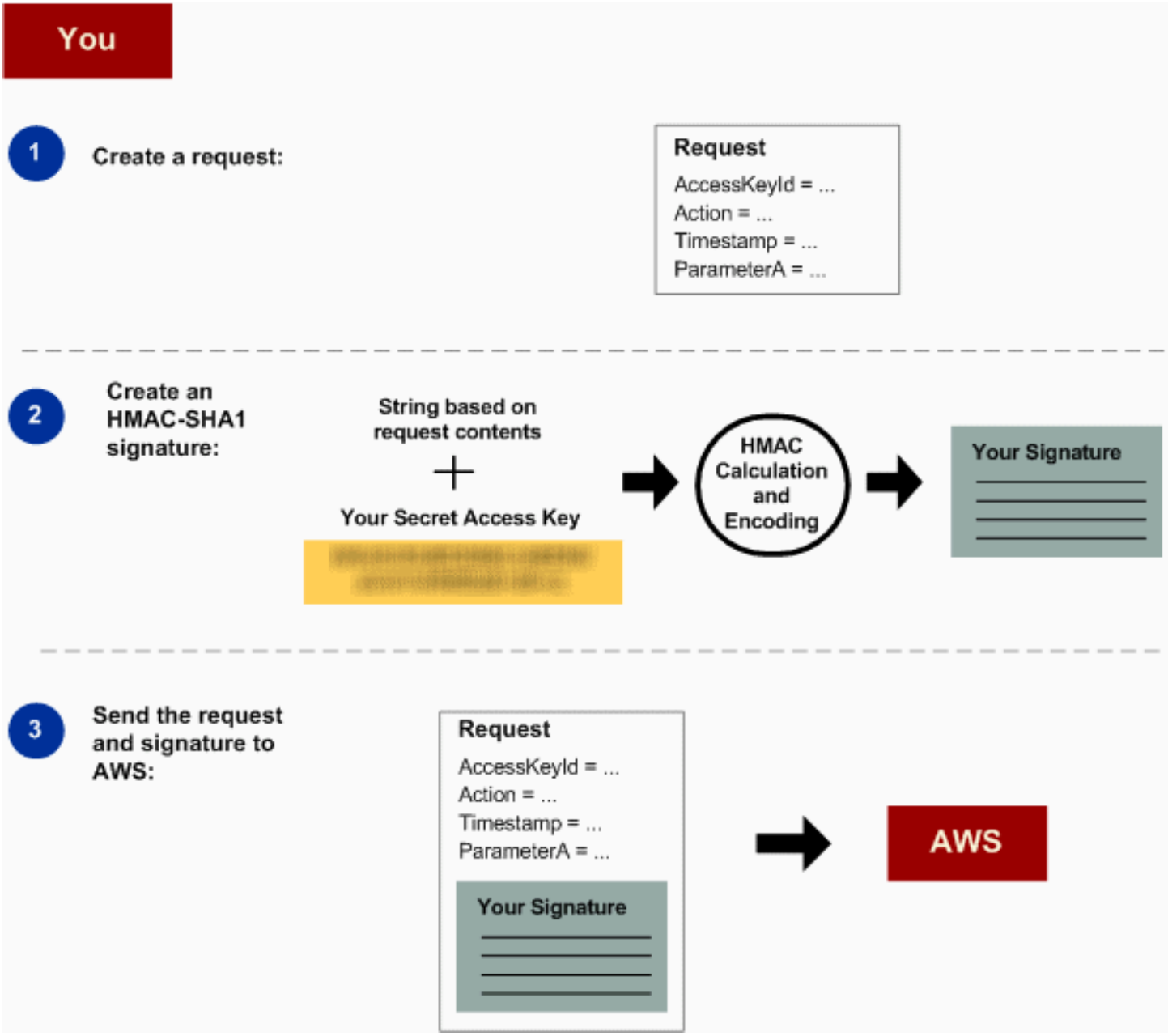
Elementos de la solicitud

- ID de clave de acceso de AWS: cada solicitud debe contener el ID de clave de acceso de la identidad que utiliza para enviar la solicitud.
- Firma: cada solicitud debe incluir una firma de solicitud válida; de lo contrario, la solicitud se rechaza.

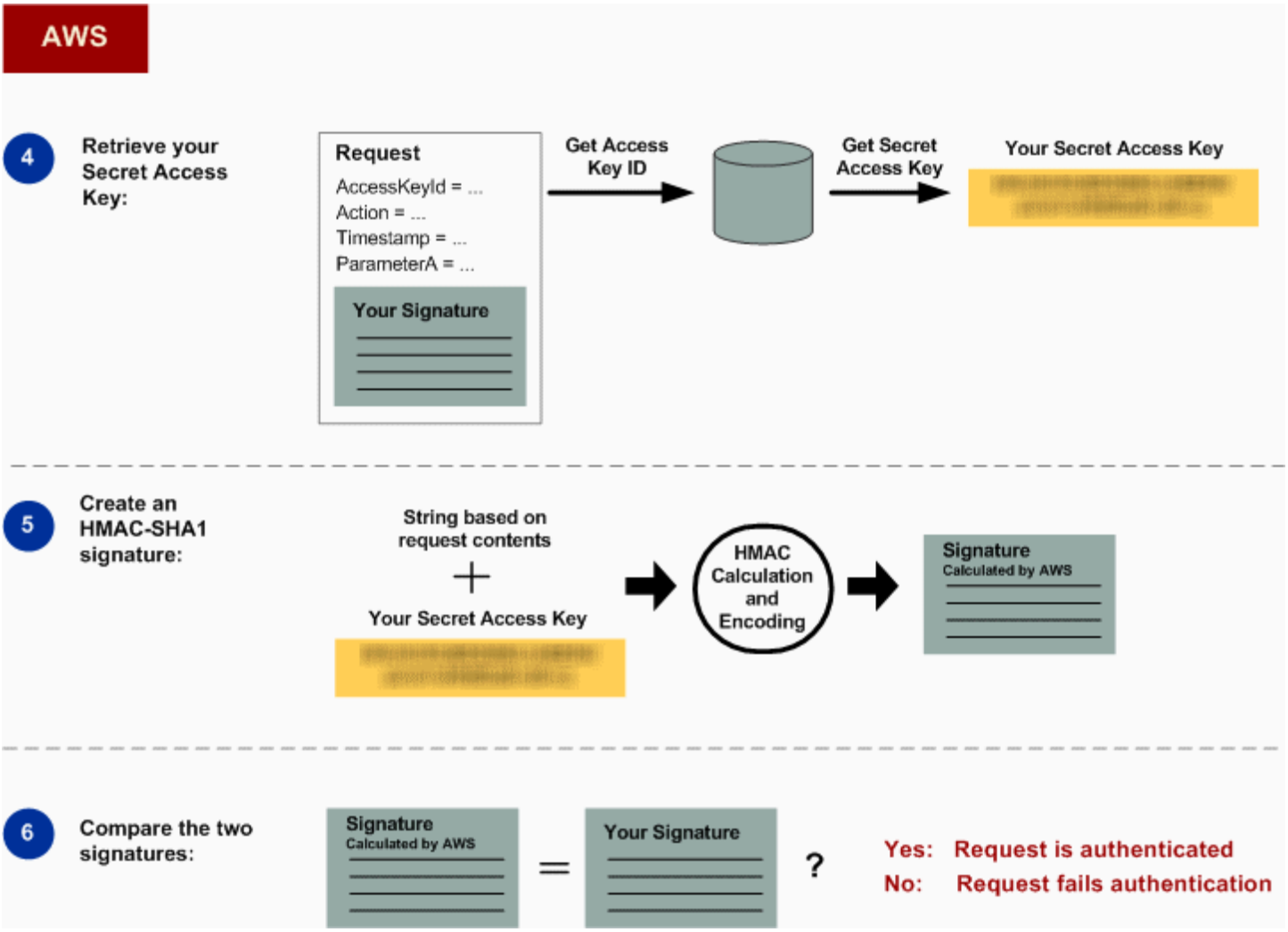
  Una firma de solicitud se calcula mediante su clave de acceso secreta, que es un secreto compartido que solo conocen usted y AWS.
- Marca temporal: cada solicitud debe incluir la fecha y la hora en que se creó la solicitud, representadas como una cadena en Universal Time Coordinated (UTC, Hora universal coordinada).
- Fecha: cada solicitud debe incluir la marca temporal de la solicitud.

  Según la acción de la API que utiliza, puede proporcionar una hora y una fecha de vencimiento para la solicitud, en lugar o además de la marca temporal. Consulte el tema de autenticación para la acción particular para determinar lo que requiere.

A continuación se indican los pasos generales para autenticar solicitudes en Amazon S3. Se asume que usted cuenta con las credenciales de seguridad, el ID de clave de acceso y la clave de acceso secreta necesarios.



1	Cree una solicitud para AWS.
2	Calcule la firma con la clave de acceso secreta.
3	Envíe la solicitud a Amazon S3. Incluya su ID de clave de acceso y la firma en su solicitud. Amazon S3 realiza los siguientes tres pasos.



4	Amazon S3 utiliza el ID de clave de acceso para buscar su clave de acceso secreta.
5	Amazon S3 calcula una firma a partir de los datos de la solicitud y la clave de acceso secreta con el mismo algoritmo que usted utilizó para calcular la firma que envió en la solicitud.
6	Si la firma generada por Amazon S3 coincide con la que envió en la solicitud, la solicitud se considera auténtica. Si la comparación falla, se descarta la solicitud y Amazon S3 devuelve una respuesta de error.

Información de autenticación detallada

Para obtener información detallada acerca de la autenticación de REST, consulte Firmar y autenticar las solicitudes REST.

Firmar y autenticar las solicitudes REST

Temas

> **Note**
>
> En este tema se explica la autenticación de solicitudes mediante el uso de Signature Versión 2. Amazon S3 ya es compatible con Signature Version 4. Esta última versión de Signature puede utilizarse en todas las regiones, y las nuevas regiones solo permitirán el uso de Signature versión 4 tras el 30 de enero de 2014. Para obtener más información, vaya a Autenticación de solicitudes (AWS Signature Version 4) en la Referencia de la API de Amazon Simple Storage Service.

La autenticación es el proceso que consiste en demostrar la identidad del usuario a un sistema. La identidad es un factor importante en las decisiones sobre el control de accesos en Amazon S3. Las solicitudes se admiten o se rechazan, en parte, en función de la identidad del solicitante. Por ejemplo, el derecho a crear buckets se reserva a los desarrolladores registrados, y el derecho a crear objetos en un bucket (de manera predeterminada) se reserva al propietario del bucket en cuestión. Como desarrollador, estará realizando solicitudes que invoquen estos privilegios, por lo que tendrá que demostrar su identidad al sistema autenticando sus solicitudes. En esta sección le demostramos cómo.

> **Note**
>
> El contenido de esta sección no se aplica al método HTTP POST. Para obtener más información, consulte Cargas basadas en el navegador con POST (AWS Signature Version 2).

La API de REST de Amazon S3 usa un esquema HTTP personalizado basado en un HMAC (Hash Message Authentication Code) con clave para la autenticación. Para autenticar una solicitud, primero ha de concatenar los elementos seleccionados en la solicitud para formar una cadena. A continuación, utilice su clave de acceso secreta de AWS para calcular el HMAC de esa cadena. Este proceso se denomina informalmente "firmar la solicitud", y al resultado del algoritmo HMAC se le llama la firma, ya que simula las propiedades de seguridad de una firma real. Por último, tendrá que agregar esta firma como parámetro de la solicitud empleando la sintaxis descrita en esta sección.

Cuando el sistema recibe una solicitud autenticada, toma la clave de acceso secreta de AWS que usted afirma tener y la utiliza del mismo modo para computar una firma para el mensaje recibido. A continuación, compara la firma calculada con la firma que presenta el solicitante. Si ambas firmas coinciden, el sistema concluye que el solicitante debe de tener acceso a la clave de acceso secreta de AWS y, por lo tanto, actúa con la autoridad de la solicitud principal para la que se emitió la clave. Si las dos firmas no coinciden, la solicitud se abandona y el sistema responde con un mensaje de error.

Example Solicitud REST autenticada de Amazon S3

```
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=
```

Uso de credenciales de seguridad temporales

Si firma su solicitud con credenciales de seguridad temporales (consulte Realizar solicitudes), deberá incluid el token de seguridad correspondiente en la solicitud agregando el encabezado `x-amz-security-token`.

Al obtener credenciales de seguridad temporales con la API de AWS Security Token Service, la respuesta incluye credenciales de seguridad temporales y un token de sesión. El valor del token de

la sesión lo facilita en el encabezado `x-amz-security-token` al enviar solicitudes a Amazon S3. Para obtener más información acerca de la API de AWS Security Token Service proporcionada por IAM, consulte la sección Acción en la Guía de referencia de la API de AWS Security Token Service.

El encabezado de autenticación

La API de REST de Amazon S3 usa el encabezado estándar HTTP `Authorization` para transmitir la información de autenticación. (El nombre "encabezado estándar" no es demasiado preciso, ya que lo que transmite es información de autenticación, no autorización). Según el esquema de autenticación de Amazon S3, el encabezado de autorización tiene la siguiente forma:

```
Authorization: AWS AWSAccessKeyId:Signature
```

Los desarrolladores reciben una ID de clave de acceso de AWS y una clave de acceso secreta de AWS cuando se registran. Para la autenticación de solicitudes, el elemento `AWSAccessKeyId` identifica la ID de clave de acceso utilizada para computar la firma e, indirectamente, también al desarrollador que realiza la solicitud.

El elemento `Signature` es el RFC 2104 HMAC-SHA1 de los elementos seleccionados de la solicitud, y por tanto, la parte `Signature` del encabezado de la autorización variará entre solicitudes. Si la firma de la solicitud calculada por el sistema coincide con la `Signature` incluida en la solicitud, el solicitante habrá demostrado la posesión de la clave de acceso secreta de AWS. La solicitud será procesada bajo la identidad y con la autoridad del desarrollador al que se le emitió la clave.

A continuación presentamos pseudogramática que ilustra la construcción del encabezado de solicitudes `Authorization`. (en el ejemplo, `\n` representa el punto de código en Unicode `U+000A`, normalmente denominado nueva línea).

```
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 es un algoritmo definido por RFC 2104, código de autenticación de mensajes en clave-hash. El algoritmo toma como entrada dos cadenas de bytes, una clave y un mensaje. Para la autenticación de solicitudes en Amazon S3, utilice su clave de acceso secreta de AWS (`YourSecretAccessKey`) como clave y la codificación UTF-8 de `StringToSign` como mensaje. El resultado del HMAC SHA1 también es una cadena de bytes, denominada resumen. El parámetro `Signature` de la solicitud se construye en Base64 mediante la codificación de este resumen.

Solicitar canonicalización para firmas

Recuerde que cuando el sistema recibe una solicitud autenticada, compara la firma de la solicitud computada con la firma proporcionada en la solicitud en `StringToSign`. Por este motivo, se debe computar la firma utilizando el mismo método que use Amazon S3. Al proceso de poner una solicitud en una forma acordada para la firma se le denomina canonicalización.

Crear elemento CanonicalizedResource

`CanonicalizedResource` representa el recurso de Amazon S3 que es el destino de la solicitud. Debe construirlo para adaptarlo a una solicitud REST de la siguiente forma:

Proceso de lanzamiento

1	Comience por una cadena vacía ("").
2	Si la solicitud especifica un bucket con el encabezado del host HTTP (estilo de alojamiento virtual), adjunte el nombre del bucket precedido por un "`/`" (por ejemplo, "/nombredelbucket"). Para las solicitudes con estilo de ruta y las solicitudes que no se dirigen a un bucket, no haga nada. Para obtener más información acerca de las solicitudes de estilo de alojamiento virtual, consulte Alojamiento virtual de buckets. Para una solicitud de tipo alojamiento virtual "https://awsexamplebucket1.s3.us-west-1.amazonaws.com/photos/puppy.jpg", el `CanonicalizedResource` es "/awsexamplebucket1". Para la solicitud de tipo ruta, "https://s3.us-west-1.amazonaws.com/awsexamplebucket1/photos/puppy.jpg", el `CanonicalizedResource` es "".

3	Adjunte la parte de la ruta del URI de la solicitud HTTP sin descodificar, hasta la cadena de la consulta, sin incluirla. Para una solicitud de tipo alojamiento virtual "https://awsexamplebucket1.s3.us-west-1.amazonaws.com/photos/puppy.jpg", el `CanonicalizedResource` es "/awsexamplebucket1/photos/puppy.jpg". Para una solicitud de tipo ruta, "https://s3.us-west-1.amazonaws.com/awsexamplebucket1/photos/puppy.jpg", el `CanonicalizedResource` es "/awsexamplebucket1/photos/puppy.jpg". En este punto, el `CanonicalizedResource` es el mismo tanto para la solicitud de estilo de alojamiento virtual como para la de estilo de ruta. Para una solicitud que no se dirija a un bucket, como GET Service, adjunte "/".
4	Si la solicitud se dirige a un subrecurso, como `?versioning` , `?location` , `?acl`, `?lifecycle` o `?versionid` , adjunte el subrecurso, su valor (si lo tiene) y el signo de interrogación. Tenga en cuenta que, en caso de que haya varios subrecursos, estos deben estar ordenados lexicográficamente por el nombre del subrecurso y separados por "&". Por ejemplo: ?acl&versionId=*`valor`*. Los subrecursos que se deben incluir cuando se crea el elemento CanonicalizedResource son los siguientes: acl, lifecycle, location, logging, notification, partNumber, policy, requestPayment, uploadId, uploads, versionId, versioning, versions y website. Si la solicitud especifica parámetros de la cadena de consulta que sobrescriban a los valores del encabezado de respuesta (véase Get Object), adjunte los parámetros de la cadena de consulta y sus valores. Al firmar no estará codificando estos valores. Sin embargo, al realizar la solicitud, debe codificar estos valores de parámetros. Los parámetros de la cadena de consulta en una solicitud GET incluyen `response-content-type` , `response-content-language` , `response-expires` , `response-cache-control` , `response-content-disposition` y `response-content-encoding` . El parámetro de la cadena de consulta `delete` debe incluirse al crear el CanonicalizedResource para una solicitud de eliminación de varios objetos.

Los elementos del CanonicalizedResource que provienen del URI de la solicitud HTTP deben firmarse literalmente según aparecen en la solicitud HTTP, incluidos los caracteres meta de codificación de la URL.

El `CanonicalizedResource` podría ser diferente del URI de la solicitud HTTP. En particular, si su solicitud usa el encabezado HTTP `Host` para especificar un bucket, el bucket no aparecerá en el URI de la solicitud HTTP. Sin embargo, el `CanonicalizedResource` seguirá incluyendo el bucket. Los parámetros de la cadena de consulta podrían aparecer también en el URI de la solicitud, pero no están incluidos en el `CanonicalizedResource`. Para obtener más información, consulte Alojamiento virtual de buckets.

Crear el elemento CanonicalizedAmzHeaders

Para construir la parte de CanonicalizedAmzHeaders de `StringToSign`, seleccione todos los encabezados de las solicitudes HTTP que comiencen por "x-amz" (con una comparación que no distinga mayúsculas y minúsculas) y emplee el siguiente proceso.

Proceso CanonicalizedAmzHeaders

1	Convierta cada nombre de encabezado HTTP a minúsculas. Por ejemplo, "`X-Amz-Date `" se ha de convertir en "`x-amz-date `".
2	Ordene la colección de encabezados lexicográficamente por nombre de encabezado.
3	Combine los campos de encabezado que tengan el mismo nombre en un par de encabezados "nombre-de-encabezado:lista-de-valores-separados-por-comas" según se indica en la RFC 2616, sección 4.2, sin espacios entre los valores. Por ejemplo, dos encabezados de metadatos "`x-amz-meta-username: fred `" y "`x-amz-meta-username: barney `" se combinarían en el encabezado único "`x-amz-meta-username: fred,barney `".
4	"Desdoble" los encabezados largos que ocupen varias líneas (según lo permite la RFC 2616, sección 4.2) sustituyendo el espacio de desdoble (incluida la línea nueva) por un espacio único.
5	Elimine los espacios que haya en torno a los dos puntos del encabezado. Por ejemplo, el encabezado "`x-amz-meta-username: fred,barney `" se convertiría en "`x-amz-meta-username:fred,barney `".
6	Por último, adjunte un nuevo carácter (`U+000A`) a cada encabezado canonicalizado de la lista resultante. Construye el elemento CanonicalizedResource concatenando todos los encabezados de esta lista en una cadena única.

Elementos StringToString de los encabezados HTTP: posicionales frente a denominados

Los primeros elementos del encabezado de `StringToSign` (Content-Type, Date y Content-MD5) tienen naturaleza posicional. `StringToSign` no incluye los nombres de estos encabezados, solo sus valores procedentes de la solicitud. Por contraste, los elementos "`x-amz-`" son denominados. Tanto los nombres como los valores de los encabezados aparecen en `StringToSign`.

Si un encabezado posicional al que se llama en la definición de `StringToSign` no está presente en su solicitud (por ejemplo, `Content-Type` o `Content-MD5` son opcionales para las solicitudes PUT y no significativos para la solicitudes GET), sustituya la cadena vacía ("") para dicha posición.

Requisitos de marca temporal

Una marca temporal válida (en la que se use el encabezado HTTP `Date` o una alternativa `x-amz-date`) es obligatoria para las solicitudes autenticadas. Además, la marca temporal del cliente incluida en una solicitud autenticada debe estar en el intervalo de 15 minutos del momento, según la hora del sistema de Amazon S3 en el que se recibe la solicitud. En caso contrario, ocurrirá un error en la solicitud y recibirá el código de error `RequestTimeTooSkewed`. El objetivo de estas restricciones es limitar la posibilidad de que las solicitudes interceptadas pudieran ser reproducidas por un adversario. Para implementar una protección más sólida ante el acceso no autorizado, use el transporte HTTPS para solicitudes autenticadas.

> **Note**
>
> La limitación de validación con la fecha de solicitud solo se aplica a las solicitudes autenticadas en las que no se use la autenticación por cadena de consulta. Para obtener más información, consulte Alternativa de autenticación por cadena de consulta de solicitudes.

Algunas bibliotecas de cliente HTTP no permiten configurar el encabezado `Date` para una solicitud. Si encuentra problemas al incluir el valor del encabezado "Date" en los encabezados canonicalizados, puede establecer la marca temporal de la solicitud mediante un encabezado "`x-amz-date`". El valor del encabezado `x-amz-date` debe encontrarse en uno de los formatos que se indican en RFC 2616 (http://www.ietf.org/rfc/rfc2616.txt). Cuando hay un encabezado `x-amz-date` presente en una solicitud, el sistema ignorará cualquier encabezado `Date` al computarla firma de la misma. Por tanto, si incluye el encabezado `x-amz-date`, use la cadena vacía para `Date` al construir el `StringToSign`. Consulte la siguiente sección para ver un ejemplo.

Ejemplos de autenticación

Los ejemplos de esta sección emplean las credenciales (no funcionales) de la siguiente tabla.

Parámetro	Valor
AWSAccessKeyId	`AKIAIOSFODNN7EXAMPLE`
AWSSecret AccessKey	`wJalrXUtnFEMI/K7MDENG/bPxRfiCYEXAMPLEKEY`

En el ejemplo, el formato de `StringToSign` no es significativo, y el `\n` es el punto de código Unicode `U+000A`, normalmente denominado línea nueva. Además, en los ejemplos se usa "+0000" para designar la zona horaria. También puede usar "GMT" para designar la zona horaria, pero las firmas que se mostrarán en los ejemplos serán diferentes.

Object GET

En este ejemplo se obtiene un objeto del bucket awsexamplebucket1.

Solicitud	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`

Tenga en cuenta que el CanonicalizedResource incluye el nombre del bucket, pero la URI de la solicitud HTTP no. (El encabezado del host especifica el bucket).

Note

La siguiente secuencia de comandos de Python calcula la firma anterior, utilizando los parámetros proporcionados. Puede utilizar este script para construir sus propias firmas, reemplazando las claves y StringToSign según corresponda.

```
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()}")
```

Object PUT

En este ejemplo se coloca un objeto en el bucket awsexamplebucket1.

Solicitud	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` `Authorization: AWS AKIAIOSFODNN7EXAMP` `LE:`	`PUT\n` `\n` `image/jpeg\n` `Tue, 27 Mar 2007 21:15:45 +0000\n` `/awsexamplebucket1/photos/puppy.jpg`

Solicitud	StringToSign
iqRzw+ileNPu1fhspnRs8nOjjIA=	

Tenga en cuenta el encabezado Content-Type en la solicitud y en el StringToSign. Tenga en cuenta también que el Content-MD5 se deja vacío en el StringToSign, porque no aparece en la solicitud.

Enumeración

En este ejemplo se muestra el contenido del bucket awsexamplebucket1.

Solicitud	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/`

Tenga en cuenta la barra al final del CanonicalizedResource y la ausencia de parámetros en la cadena de consulta.

Fetch

En este ejemplo se obtiene el subrecurso de la política de control de acceso para el bucket "awsexamplebucket1".

Solicitud	StringToSign
`GET /?acl HTTP/1.1` `Host: awsexamplebucket1.s3.us-west-1.amazonaws.com`	`GET\n` `\n` `\n`

Solicitud	StringToSign
`Date: Tue, 27 Mar 2007 19:44:46 +0000` `Authorization: AWS AKIAIOSFODNN7EXAMPLE:` `82ZHiFIjc+WbcwFKGUVEQspPn+0=`	`Tue, 27 Mar 2007 19:44:46` `+0000\n` `/awsexamplebucket1/?acl`

Ahora, tenga en cuenta cómo el parámetro de la cadena de consulta del subrecurso está incluido en el CanonicalizedResource.

Eliminación

En este ejemplo se elimina un objeto del bucket "awsexamplebucket1" de tipo ruta y la alternativa Date.

Solicitud	StringToSign
`DELETE /awsexamplebucket1/photos/p` `uppy.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 AKIAIOSFODNN7EXAMP` `LE:XbyTlbQdu9Xw5o8P4iMwPktxQd8=`	`DELETE\n` `\n` `\n` `Tue, 27 Mar 2007 21:20:26 +0000\n` `/awsexamplebucket1/photos/puppy.jpg`

Tenga en cuenta que hemos usado el método alternativo "x-amz-date" para especificar la fecha (porque nuestra biblioteca de clientes, por ejemplo, nos impide establecer la fecha). En este caso, el `x-amz-date` tiene prioridad sobre el encabezado `Date`. Por tanto, la fecha introducida en la firma ha de contener el valor del encabezado `x-amz-date`.

Cargar

En este ejemplo se carga un objeto en un bucket con alojamiento virtual de estilo CNAME y con metadatos.

Solicitud	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
```

Tenga en cuenta cómo los encabezados "x-amz-" se ordenan, se les recortan los espacios extra o se convierten en minúscula. Tenga en cuenta también que se han agrupado varios encabezados con el mismo nombre utilizando comas para separar valores.

Tenga en cuenta que solo los encabezados de entidades HTTP `Content-Type` y `Content-MD5` aparecen en el `StringToSign`. El resto de encabezados de entidades `Content-*` no aparecen.

Además, tenga en cuenta que el `CanonicalizedResource` incluye el nombre del bucket, pero el URI de la solicitud HTTP no lo incluye. (El encabezado del host especifica el bucket).

Mostrar todos mis buckets

Solicitud	StringToSign

```
GET / HTTP/1.1
```

```
GET\n
```

Solicitud	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` `/`

Claves de Unicode

Solicitud	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**
>
> Los elementos en el `StringToSign` que se derivaron del URI de la solicitud se toman literalmente, incluida la codificación y capitalización de la URL.

Problemas de firma con solicitudes REST

Cuando falla la autenticación de REST, el sistema responde a la solicitud con un documento de errores XML. La información que contiene este documento de errores tiene como objetivo ayudar a los desarrolladores a diagnosticar el problema. En particular, el elemento `StringToSign` del documento de errores `SignatureDoesNotMatch` le dice exactamente qué canonicalización de solicitudes está empleando el sistema.

Algunas herramientas insertan encabezados en modo silencioso cuya existencia usted no conocía, como la agregación del encabezado `Content-Type` durante un PUT. En la mayoría de estos casos, el valor del encabezado insertado permanece constante, con lo que podrá descubrir los encabezados faltantes con herramientas como Ethereal o tcpmon.

Alternativa de autenticación por cadena de consulta de solicitudes

Puede autenticar determinado tipo de solicitudes pasando la información requerida como parámetros de una cadena de consulta, en lugar de usar el encabezado HTTP `Authorization`. Esto resulta útil para habilitar el acceso directo por navegador de terceros a sus datos de Amazon S3 privados sin hacer pasar la solicitud por un proxy. La idea es construir una solicitud "prefirmada" y codificarla como una URL que pueda recuperar el navegador de un usuario final. Además, puede limitar una solicitud prefirmada especificando una fecha de vencimiento.

Para obtener más información acerca del uso de parámetros de consulta para autenticar solicitudes, consulte Autenticación de solicitudes: Uso de los parámetros de consulta(AWS Signature Version 4) en la Referencia de la API de Amazon Simple Storage Service. Con el fin de ver ejemplos sobre cómo usar los SDK de AWS para generar URL prefirmadas, consulte Uso compartido de objetos con URL prefirmadas.

Crear una firma

A continuación figura un ejemplo de una solicitud REST de Amazon S3 autenticada por cadena de consulta.

```
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
```

El método de autenticación de solicitudes con cadena de consulta no requiere ningún encabezado HTTP especial. Los elementos de autenticación necesarios se especifican como parámetros de la cadena de consulta:

Nombre del parámetro de la cadena de consulta	Ejemplo de valor	Descripción
`AWSAccessKeyId`	`AKIAIOSFODNN7EXAMPLE`	El ID de la clave de acceso de AWS. Especifica la clave de acceso secreta de AWS utilizada para firmar la solicitud e, indirectamente, la

Nombre del parámetro de la cadena de consulta	Ejemplo de valor	Descripción
		identidad del desarrollador que realiza la solicitud.
`Expires`	`1141889120`	El momento de vencimiento de la firma, especificado como el número de segundos a partir de la fecha de inicio (00:00:00 UTC del 1 de enero de 1970). Cualquier solicitud recibida con posterioridad a este momento (según el servidor) será rechazada.
`Signature`	`vjbyPxybdZaNmGa%2B yT272YEAiv4%3D`	La codificación URL de la codificación Base64 del HMAC SHA1 de StringToSign.

El método de autenticación de solicitudes con cadena de consulta difiere levemente del método ordinario, pero solo en el formato del parámetro de la solicitud `Signature` y el elemento `StringToSign`. A continuación presentamos pseudogramática que ilustra el método de autenticación de solicitudes por cadena de consulta.

```
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` es el ID de clave de acceso secreta de AWS que le asigna Amazon cuando se registra para ser un desarrollador de Amazon Web Services. Tenga en cuenta que la

`Signature` está codificada en formato de URL para que se pueda colocar en la cadena de consulta. Tenga en cuenta también que, en `StringToSign`, el elemento posicional HTTP `Date` ha sido sustituido por `Expires`. El `CanonicalizedAmzHeaders` y el `CanonicalizedResource` son iguales.

> **Note**
>
> En el método de autenticación por cadena de consulta, no utilice el encabezado `Date` ni `x-amz-date request` al calcular la cadena para firmar.

Autenticación por cadena de consulta de solicitudes

Solicitud	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`

Suponemos que, cuando un navegador realiza la solicitud GET, no facilitará un encabezado Content-MD5 ni Content-Type, ni tampoco creará encabezados x-amz-, por lo que esas partes de `StringToSign` se dejan vacías.

Usar codificación Base64

Las firmas de solicitudes HMAC deben tener codificación Base64. La codificación Base64 convierte la firma en una cadena ASCII sencilla que se puede adjuntar a la solicitud. Los caracteres que podrían aparecer en la cadena de firma, como más (+), barra inclinada (/) e igual (=) deben estar codificados si se usan en un URI. Por ejemplo, si el código de autenticación incluye un signo más (+), codifíquelo como %2B en la solicitud. Las barras inclinadas se codifican como %2F, y los signos de igual, como %3D.

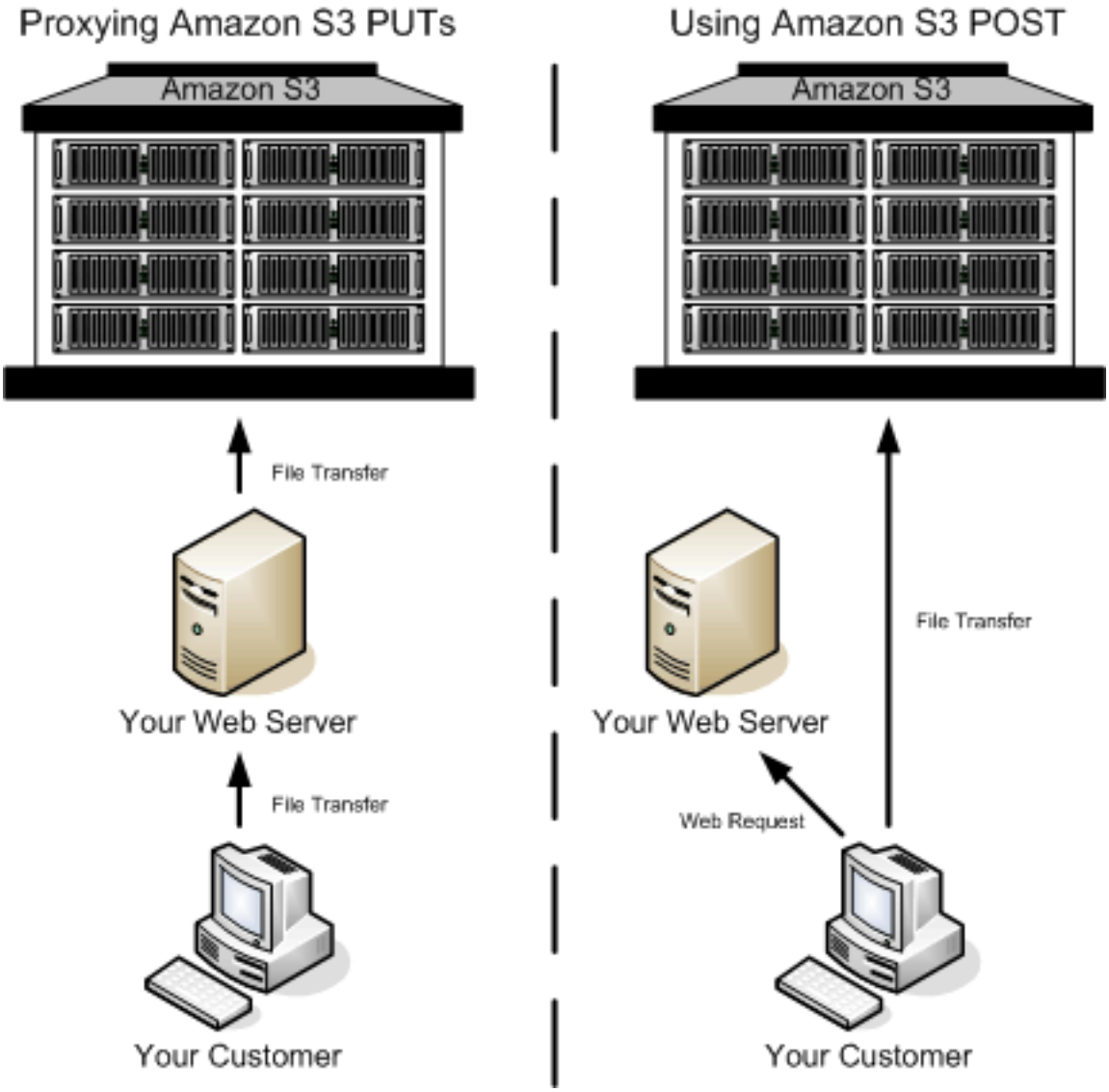
Para ver más ejemplos de codificación en Base64, consulte los de Amazon 3 Ejemplos de autenticación.

Cargas basadas en el navegador con POST (AWS Signature Version 2)

Amazon S3 admite POST, que le permite a los usuarios cargar contenido directamente en Amazon S3. POST está diseñado para simplificar las cargas, reducir la latencia de las cargas y ahorrar dinero en aplicaciones que los usuarios utilizan para cargar datos y guardarlos en Amazon S3.

> **Note**
>
> La autenticación de solicitudes que se analiza en esta sección se basa en AWS Signature Version 2, un protocolo para autenticar las solicitudes de la API entrantes para los servicios de AWS.
>
> Amazon S3 ahora admite Signature Version 4, un protocolo para autenticar las solicitudes de la API entrantes para los servicios de AWS, en todas las Regiones de AWS. En este momento, las Regiones de AWS creadas antes del 30 de enero de 2014 seguirán admitiendo el protocolo anterior, Signature Version 2. Cualquier región nueva después del 30 de enero de 2014 solo admitirá Signature Version 4, y por lo tanto, todas las solicitudes de esas regiones se deben realizar con Signature Version 4. Para obtener más información, consulte Autenticación de solicitudes de las cargas basadas en el navegador con POST (AWS Signature Version 4) en la Referencia de la API de Amazon Simple Storage Service.

En la siguiente figura se muestra una carga con POST de Amazon S3.



Cargar con POST

1	El usuario abre un navegador web y obtiene acceso a su página web.
2	Su página web incluye un formulario HTTP que contiene toda la información necesaria para que el usuario pueda cargar contenido en Amazon S3.
3	El usuario carga contenido directamente en Amazon S3.

> **Note**
>
> POST no admite la autenticación por query string.

Formularios HTML (AWS Signature Version 2)

Temas

Cuando se comunica con Amazon S3, por lo general utiliza la API REST o SOAP para realizar operaciones como PUT, GET, DELETE, y otras. Con POST, los usuarios cargan datos directamente en Amazon S3 a través de sus navegadores, que no pueden procesar la API de SOAP ni crear una solicitud PUT de REST.

> **Note**
>
> La compatibilidad con SOAP por HTTP está obsoleta, pero SOAP aún se encuentra disponible con HTTPS. Las características nuevas de Amazon S3 no son compatibles con SOAP. En vez de usar SOAP, le recomendamos que utilice la API de REST o los SDK de AWS.

Para que los usuarios puedan cargar contenido en Amazon S3 con sus navegadores, debe utilizar los formularios HTML. Los formularios HTML constan de una declaración de formulario y campos de formulario. Cada declaración de formulario incluye información de alto nivel acerca de la solicitud. Los campos de formulario incluyen información detallada acerca de la solicitud, así como la política que se utiliza para autenticarla y asegurar que cumpla con las condiciones que usted especifica.

> **Note**
>
> Los datos y límites del formulario (sin incluir los contenidos del archivo) no pueden exceder los 20 KB.

En esta sección se explica cómo utilizar los formularios HTML.

Codificar formulario HTML

El formulario y la política deben estar cifrados con UTF-8. Para aplicar la codificación UTF-8 en el formulario puede especificarlo en el encabezado HTML o como un encabezado de solicitud.

> **Note**
>
> La declaración de formulario HTML no acepta parámetros de autenticación por query string.

A continuación, mostramos un ejemplo de la codificación UTF-8 en el encabezado HTML:

```
<html>
  <head>
    ...
    <meta http-equiv="Content-Type" content="text/html; charset=UTF-8" />
    ...
  </head>
  <body>
```

A continuación, se muestra un ejemplo de la codificación UTF-8 en un encabezado de solicitud:

```
Content-Type: text/html; charset=UTF-8
```

Declaración de formulario HTML

La declaración de formulario tiene tres componentes: la acción, el método y el tipo de documento adjunto. Si cualquiera de estos valores se configura de manera inadecuada, la solicitud falla.

La acción especifica el URL que procesa la solicitud, que debe establecerse en el URL del bucket. Por ejemplo, si el nombre de su bucket es `awsexamplebucket1` y la región es EE. UU. Oeste (Norte de California), la URL es `https://awsexamplebucket1.s3.us-west-1.amazonaws.com/`.

> **Note**
>
> El nombre de clave se especifica en un campo de formulario.

El método debe ser POST.

Se debe especificar el tipo de documento adjunto (enctype) y se debe establecer en datos de formulario/multiparte para cargas de archivos y cargas de área de texto. Para obtener más información, visite RFC 1867.

Example

El siguiente ejemplo es una declaración de formulario para el bucket "awsexamplebucket1".

```
<form action="https://awsexamplebucket1.s3.us-west-1.amazonaws.com/" method="post"

enctype="multipart/form-data">
```

Campos de formulario HTML

En la siguiente tabla se describen los campos que se pueden utilizar en un formulario HTML.

> **Note**
>
> La variable `${filename}` se reemplaza automáticamente con el nombre del archivo provisto por el usuario y es reconocida por todos los campos de formulario. Si el navegador o el cliente proporciona una ruta total o parcial al archivo, solo se utilizará el texto después de la última barra inclinada (/) o barra inversa (\). Por ejemplo, “C:\Program Files\directory1\file.txt” se interpretará como “file.txt”. Si no se brinda ningún archivo o nombre de archivo, la variable se reemplaza con una cadena vacía.

Nombre del campo	Descripción	Obligatorio
`AWSAccessKeyId`	El ID de clave de acceso de AWS del propietario del bucket que otorga un acceso de usuario anónimo para una solicitud que cumple con el conjunto de restricciones en la política. Este campo es obligatorio si la solicitud incluye un documento de política.	Condicional
`acl`		No

Nombre del campo	Descripción	Obligatorio
	Una lista de control de acceso (ACL) de Amazon S3. Si se especifica una lista de control de acceso no válida, se genera un error. Para obtener más información acerca de las ACL, consulte Listas de control de acceso (ACL). Tipo: String Valor predeterminado: privado Valores válidos: `private | public-read | public-read-write | aws-exec-read | authenticated-read | bucket-owner-read | bucket-owner-full-control`	
`Cache-Control, Content-Type, Content-Disposition, Content-Encoding, Expires`	Encabezados específicos de REST. Para obtener más información, consulte PUT Object.	No
`key`	El nombre de la clave cargada. Para utilizar el nombre de archivo provisto por el usuario, utilice la variable ${filename}. Por ejemplo si la usuaria Betty carga el archivo lolcatz.jpg y usted especifica /user/betty/${filename}, el archivo se guarda como /user/betty/lolcatz.jpg. Para obtener más información, consulte Trabajar con metadatos de objeto.	Sí

Nombre del campo	Descripción	Obligatorio
`policy`	Política de seguridad que describe qué está permitido en la solicitud. Las solicitudes sin una política de seguridad se consideran anónimas y solo se aceptarán en buckets que se pueden escribir públicamente.	No
`success_action_redirect, redirect`	URL al que el cliente es redirigido después de la carga exitosa. Amazon S3 adjunta al URL el bucket, la clave y los valores de la etag como parámetros de cadena de consulta. Si no se especifica el campo success_action_redirect, Amazon S3 devuelve el tipo de documento vacío especificado en el campo success_action_status. Si Amazon S3 no puede interpretar la URL, ignora el campo . Si la carga falla, Amazon S3 muestra un error y no redirige al usuario a una URL. Para obtener más información, consulte Redireccionamiento. **Note** El nombre de campo de redirección es obsoleto y el soporte para el nombre de campo de redirección se eliminará en el futuro.	No

Nombre del campo	Descripción	Obligatorio
`success_action_status`	El código de estado que recibe el cliente después de la carga exitosa si no se especific a el campo success_action_redirect. Los valores válidos son 200, 201 o 204 (predeterminado). Si el valor se establece en 200 o 204, Amazon S3 devuelve un documento vacío con un código de estado 200 o 204. Si el valor se establece en 201, Amazon S3 devuelve un documento XML con un código de estado 201. Para obtener información acerca del contenido del documento XML, consulte POST Object. Si el valor no se establece o si se establece en un valor no válido, Amazon S3 devuelve un documento vacío con un código de estado 204. **Note** Algunas versiones de Adobe Flash Player no controlan debidamente las respuestas HTTP con un cuerpo vacío. Para admitir cargas con Adobe Flash, recomendamos establecer `success_action_status` en 201.	No

Nombre del campo	Descripción	Obligatorio
`signature`	La firma HMAC que se crea con la clave de acceso secreta que corresponde al AWSAccessKeyId provisto. Este campo es obligatorio si se incluye un documento de política en la solicitud. Para obtener más información, consulte Administración de identidades y accesos para Amazon S3.	Condicional
`x-amz-security-token`	Un token de seguridad utilizado por las credenciales de sesión Si la solicitud utiliza Amazon DevPay, se requieren dos campos de formulario `x-amz-security-token` : uno para el token de producto y otro para el token de usuario. Si la solicitud utiliza credenciales de sesión, se requiere un formulario `x-amz-security-token` . Para obtener más información, consulte Credenciales de seguridad temporales en la guía del usuario de IAM.	No
Otros nombres de archivo con prefijos x-amz-meta-	Metadatos especificados por el usuario. Amazon S3 no valida ni utiliza estos datos. Para obtener más información, consulte PUT Object.	No

Nombre del campo	Descripción	Obligatorio
file	Contenido de texto o archivo. El archivo o el contenido debe ser el último campo en el formulario. Cualquier campo debajo de estos se ignora. No puede cargar más de un archivo a la vez.	Sí

Construcción de la política

Temas

- Expiration
- Condiciones
- Coincidencia de condiciones
- Secuencia de escape de caracteres

La política es un documento JSON con codificación UTF-8 y Base64 que especifica las condiciones que debe cumplir la solicitud, y se utiliza para autenticar el contenido. Según cómo diseñe sus documentos de política, puede utilizarlos por carga, por usuario, para todas las cargas o de acuerdo con otros diseños que se ajusten a sus necesidades.

> **Note**
>
> Si bien el documento de política es opcional, lo recomendamos ampliamente en lugar de hacer un bucket que se pueda escribir públicamente.

A continuación se muestra el ejemplo de un documento de política:

```
{ "expiration": "2007-12-01T12:00:00.000Z",

  "conditions": [

    {"acl": "public-read" },
```

```
    {"bucket": "awsexamplebucket1" },
    ["starts-with", "$key", "user/eric/"],
  ]
}
```

El documento de política incluye los vencimientos y las condiciones.

Expiration

El elemento de vencimiento especifica la fecha de vencimiento de la política en el formato de fecha según la norma ISO 8601 en Universal Time Coordinated (UTC, Hora universal coordinada). Por ejemplo, “2007-12-01T12:00:00.000Z” especifica que la política no es válida después de la medianoche, UTC, del 01/12/2007. El vencimiento es obligatorio en una política.

Condiciones

Las condiciones en el documento de política validan los contenidos del objeto cargado. Cada campo de formulario que especifica en el formulario (salvo AWSAccessKeyId, firma, archivo, política y nombres de archivos que tienen un prefijo x-ignore-) se debe incluir en la lista de condiciones.

> **Note**
>
> Si tiene varios campos con el mismo nombre, los valores deben estar separados por comas. Por ejemplo, si tiene dos campos denominados “x-amz-meta-tag” y el primero tiene el valor “Ninja” y el segundo tiene el valor “Stallman”, configurará el documento de política como `Ninja,Stallman`.
>
> Todas las variables en el formulario se expanden antes de que la política se valide. Por lo tanto, se deben realizar todas las coincidencias de condiciones con respecto a los campos expandidos. Por ejemplo, si configuró el campo de clave en `user/betty/${filename}`, su política puede ser `[ "starts-with", "$key", "user/betty/" ]`. No escriba `[ "starts-with", "$key", "user/betty/${filename}" ]`. Para obtener más información, consulte Coincidencia de condiciones.

En la siguiente tabla se describen las condiciones de los documentos de política.

Nombre del elemento	Descripción
acl	Especifica las condiciones que debe cumplir la ACL. Admite las coincidencias exactas y `starts-with` .
content-length-range	Especifica el tamaño mínimo y máximo permitido para el contenido cargado. Admite la coincidencia de rango.
Cache-Control, Content-Type, Content-Disposition, Content-Encoding, Expires	Encabezados específicos de REST. Admite las coincidencias exactas y `starts-with` .
clave	El nombre de la clave cargada. Admite las coincidencias exactas y `starts-with` .
success_action_redirect, redirect	URL al que el cliente es redirigido después de la carga exitosa. Admite las coincidencias exactas y `starts-with` .
success_action_status	El código de estado que recibe el cliente después de la carga exitosa si no se especifica el campo success_action_redirect. Admite las coincidencias exactas.
x-amz-security-token	Token de seguridad de Amazon DevPay. Cada solicitud que utiliza Amazon DevPay requiere dos campos de formulario `x-amz-security-token` : uno para el token de producto y otro para el token de usuario. Como consecuencia, los valores deben estar separados por comas. Por ejemplo, si el token de usuario es `eW91dHViZ`

Nombre del elemento	Descripción
	`Q==` y el token del producto es `b0hnNVNKWVJIQTA=` , establece la entrada de política en `{ "x-amz-security-token": "eW91dHViZQ==,b0hnNVNKWVJIQTA=" }` .
Otros nombres de archivo con prefijos x-amz-meta-	Metadatos especificados por el usuario. Admite las coincidencias exactas y `starts-with` .

> **Note**
>
> Si su conjunto de herramientas añade campos adicionales (p. ej., Flash añade el nombre de archivo), debe añadirlos al documento de política. Si puede controlar esta funcionalidad, añada el prefijo `x-ignore-` al campo para que Amazon S3 omita la característica y no afecte futuras versiones de esta característica.

Coincidencia de condiciones

En la siguiente tabla se describen los tipos de coincidencias de condiciones. Si bien debe especificar una condición para cada campo de formulario que especifica en el formulario, puede crear criterios de coincidencia más complejos especificando varias condiciones para un campo de formulario.

Condición	Descripción
Coincidencias exactas	Las coincidencias exactas verifican que los campos coincidan con valores específicos. En este ejemplo se indica que la ACL se debe establecer en public-read: `{"acl": "public-read" }` Este ejemplo es una alternativa para indicar que la ACL se debe establecer en public-read:

Condición	Descripción
	`[ "eq", "$acl", "public-read" ]`
Empieza por	Si el valor debe empezar con un valor determinado, utilice starts-with. En este ejemplo se indica que la clave debe empezar con user/betty: `["starts-with", "$key", "user/betty/"]`
Coincidencia con cualquier contenido	Para configurar la política para permitir cualquier contenido dentro de un campo, utilice starts-with con un valor vacío. Este ejemplo permite cualquier campo success_action_redirect: `["starts-with", "$success_action_redirect", ""]`
Especificación de rangos	Para campos que aceptan rangos, separe los rangos superiores e inferiores con una coma. Este ejemplo permite un tamaño de archivo de 1 a 10 megabytes: `["content-length-range", 1048579, 10485760]`

Secuencia de escape de caracteres

En la siguiente tabla se describen los caracteres a los que se debe aplicar una secuencia de escape en un documento de política.

Secuencia de escape	Descripción
\\	Barra inversa
\$	Signo de dólar

Secuencia de escape	Descripción
\b	Retroceso
\f	Salto de página
\n	Nueva línea
\r	Salto de línea
\t	Tabulador horizontal
\v	Tabulador vertical
\u*xxxx*	Todos los caracteres Unicode

Crear una firma

Paso	Descripción
1	Codifique la política con UTF-8.
2	Codifique los bytes de UTF-8 con Base64.
3	Firme la política con su clave de acceso secreta mediante el uso del algoritmo HMAC SHA-1.
4	Codifique la firma de SHA-1 con Base64.

Para obtener más información acerca de la autenticación, consulte Administración de identidades y accesos para Amazon S3.

Redireccionamiento

En esta sección se describe cómo administrar los direccionamientos.

Redireccionamiento general

Al finalizar la solicitud POST, el usuario es redirigido a la ubicación que especificó en el campo `success_action_redirect`. Si Amazon S3 no puede interpretar la URL, ignora el campo `success_action_redirect`.

Si no se especifica el campo `success_action_redirect`, Amazon S3 devuelve el tipo de documento vacío especificado en el campo `success_action_status`.

Si la solicitud POST falla, Amazon S3 muestra un error y no proporciona un redireccionamiento.

Redireccionamiento de carga previa

Si su bucket se creó con <CreateBucketConfiguration>, sus usuarios finales pueden necesitar una redirección. Si esto sucede, algunos navegadores pueden administrar la redirección de manera incorrecta. Esto es relativamente poco frecuente, pero es muy probable que suceda inmediatamente después de que se crea un bucket.

Ejemplos de carga (AWS Signature Version 2)

Temas

- Cargar archivo
- Cargar área de texto

> **Note**
>
> La autenticación de solicitudes que se analiza en esta sección se basa en AWS Signature Version 2, un protocolo para autenticar las solicitudes de la API entrantes para los servicios de AWS.
> Amazon S3 ahora admite Signature Version 4, un protocolo para autenticar las solicitudes de la API entrantes para los servicios de AWS, en todas las Regiones de AWS. En este momento, las Regiones de AWS creadas antes del 30 de enero de 2014 seguirán admitiendo

el protocolo anterior, Signature Version 2. Cualquier región nueva después del 30 de enero de 2014 solo admitirá Signature Version 4, y por lo tanto, todas las solicitudes de esas regiones se deben realizar con Signature Version 4. Para obtener más información, consulte Ejemplos de cargas basadas en el navegador con HTTP POST (mediante AWS Signature Version 4) en la Referencia de la API de Amazon Simple Storage Service.

Cargar archivo

En este ejemplo se muestra el proceso completo para crear una política y un formulario que se puede utilizar para cargar un archivo adjunto.

Crear política y formulario

La siguiente política admite cargas en Amazon S3 para el bucket 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", ""]
  ]
}
```

Esta política requiere lo siguiente:

- La carga se debe realizar el 1 de diciembre de 2007 antes de las 12:00 UTC.
- El contenido se debe cargar en el bucket awsexamplebucket1.
- La clave debe empezar con “user/eric/”.
- La ACL se establece en public-read.
- El campo success_action_redirect se establece en https://awsexamplebucket1.s3.us-west-1.amazonaws.com/successful_upload.html.
- El objeto es un archivo de imagen.

- La etiqueta x-amz-meta-uuid se debe establecer en 14365123651274.
- El campo x-amz-meta-tag puede incluir cualquier valor.

A continuación, se incluye una versión de esta política codificada en Base64.

```
eyAiZXhwaXJhdGlvbiI6ICIyMDA3LTEyLTAxVDEyOjAwOjAwLjAwMFoiLAogICJjb25kaXRpb25zIjogWwogICAgeyJidWN
```

Con su credenciales, cree una firma, por ejemplo `0RavWzkygo6QX9caELEqKi9kDbU=` es la firma para el documento de política anterior.

El siguiente formulario admite una solicitud POST al bucket amzn-s3-demo-bucket que utiliza esta política.

```
<html>
  <head>
    ...
    <meta http-equiv="Content-Type" content="text/html; charset=UTF-8" />
    ...
  </head>
  <body>
  ...
  <form action="https://amzn-s3-demo-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>
```

Solicitar ejemplo

Esta solicitud asume que la imagen cargada tiene una tamaño de 117 108 bytes; los datos de la imagen no se incluyen.

```
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--
```

Respuesta de ejemplo

```
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
```

Cargar área de texto

Temas

- Crear política y formulario
- Solicitar ejemplo
- Respuesta de ejemplo

En el siguiente ejemplo se muestra el proceso completo para crear una política y un formulario para cargar un área de texto. Cargar un área de texto es útil para presentar contenido creado por el usuario, como publicaciones de blog.

Crear política y formulario

La siguiente política admite cargas de área de texto en Amazon S3 para el bucket 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", ""]
  ]
}
```

Esta política requiere lo siguiente:

- La carga se debe realizar antes de las 12:00 GMT del 01/12/2007.
- El contenido se debe cargar en el bucket awsexamplebucket1.
- La clave debe empezar con “user/eric/”.
- La ACL se establece en public-read.
- El campo success_action_redirect se establece en https://awsexamplebucket1.s3.us-west-1.amazonaws.com/new_post.html.
- El objeto es un texto HTML.
- La etiqueta x-amz-meta-uuid se debe establecer en 14365123651274.
- El campo x-amz-meta-tag puede incluir cualquier valor.

A continuación, se incluye una versión de esta política codificada en Base64.

```
eyAiZXhwaXJhdGlvbiI6ICIyMDA3LTEyLTAxVDEyOjAwOjAwLjAwMFoiLAogICJjb25kaXR
```

```
pb25zIjogWwogICAgeyJidWNrZXQiOiAiam9obnNtaXRoIn0sCiAgICBbInN0YXJ0cy13aXRoIiwgIiRrZXkiLCAidXNlci
LAogICAgeyJhY2wiOiAicHVibGljLXJlYWQifSwKICAgIHsic3VjY2Vzc19hY3Rpb25fcmVkaXJlY3QiOiAiaHR0cDovL2p
C5zMy5hbWF6b25hd3MuY29tL25ld19wb3N0Lmh0bWwifSwKICAgIFsiZXEiLCAiJENvbnRlbnQtVHlwZSIsICJ0ZXh0L2h0
CAgIHsieC1hbXotbWV0YS11dWlkIjogIjE0MzY1MTIzNjUxMjc0In0sCiAgICBbInN0YXJ0cy13aXRoIiwgIiR4LWFtei1t
IsICIiXQogIF0KfQo=
```

Con sus credenciales, cree una firma. Por ejemplo, qA7FWXKq6VvU68lI9KdveT1cWgF= es la firma para el documento de política anterior.

El siguiente formulario admite una solicitud POST al bucket amzn-s3-demo-bucket que utiliza esta política.

```
<html>
  <head>
    ...
    <meta http-equiv="Content-Type" content="text/html; charset=UTF-8" />
    ...
  </head>
  <body>
  ...
  <form action="https://amzn-s3-demo-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>
```

Solicitar ejemplo

Esta solicitud asume que la imagen cargada tiene una tamaño de 117 108 bytes; los datos de la imagen no se incluyen.

```
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--
```

Respuesta de ejemplo

```
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 con Adobe Flash

En esta sección se describe cómo utilizar POST con Adobe Flash.

Seguridad de Adobe Flash Player

De forma predeterminada, el modelo de seguridad de Adobe Flash Player les prohíbe a los usuarios de Adobe Flash Player conectarse a servidores fuera del domino al que sirve el archivo SWF.

Para sobrescribir el valor predeterminado, debe cargar un archivo crossdomain.xml que se pueda leer públicamente al bucket que aceptará cargas POST. A continuación se muestra un archivo crossdomain.xml de ejemplo.

```
<?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**
>
> Para obtener más información acerca del modelo de seguridad de Adobe Flash, visite el sitio web de Adobe.
> La incorporación del archivo crossdomain.xml a su bucket permite que cualquier usuario de Adobe Flash Player se conecte al archivo crossdomain.xml en su bucket; sin embargo, esto no brinda acceso al bucket de Amazon S3 real.

Consideraciones sobre Adobe Flash

La API FileReference en Adobe Flash añade el campo de formulario `Filename` a la solicitud POST. Al crear aplicaciones de Adobe Flash que se cargan en Amazon S3 con la acción de la API FileReference, incluya la siguiente condición en su política:

```
['starts-with', '$Filename', '']
```

Algunas versiones de Adobe Flash Player no controlan debidamente las respuestas HTTP que tienen un cuerpo vacío. Para configurar POST de manera que devuelva una respuesta que no tenga un cuerpo vacío, establezca el campo `success_action_status` en 201. Amazon S3 devolverá un documento XML con un código de estado 201. Para obtener información acerca del contenido del documento XML, consulte POST Object. Para obtener información acerca de los campos del formulario, consulte Campos de formulario HTML.

Prácticas recomendadas para patrones de diseño: optimizar el rendimiento de Amazon S3

Sus aplicaciones pueden lograr fácilmente miles de transacciones por segundo en el rendimiento de la solicitud al cargar y recuperar almacenamiento desde Amazon S3. Amazon S3 se escala automáticamente a velocidades de solicitudes altas. Por ejemplo, la aplicación puede realizar al menos 3500 solicitudes PUT/COPY/POST/DELETE o 5500 solicitudes GET/HEAD por segundo y prefijo de Amazon S3 dividido. No existe ningún límite en cuanto al número de prefijos dentro de un bucket. Para aumentar el rendimiento de lectura o escritura, ejecute en paralelo las operaciones de lectura. Por ejemplo, si crea 10 prefijos en un bucket de Amazon S3 para ejecutar en paralelo las operaciones de lectura, podría escalar el rendimiento de lectura a 55 000 solicitudes de lectura por segundo. Del mismo modo, puede escalar las operaciones de escritura escribiendo en varios prefijos. La escalación, en el caso de las operaciones de lectura y escritura, se produce gradualmente y no es instantánea. A medida que Amazon S3 escala según la nueva tasa de solicitudes más elevada, es posible que aparezcan algunos errores 503 (ralentización). Estos errores desaparecerán cuando se complete la escalación. Para obtener más información acerca de cómo crear y utilizar prefijos, consulte Organizar objetos con prefijos.

Por ejemplo, algunas aplicaciones de lago de datos de Amazon S3 analizan millones o miles de millones de objetos para consultas que ejecutan petabytes de datos. Estas aplicaciones de lagos de datos logran velocidades de transferencia de una sola instancia que maximizan el uso de la interfaz de red para su instancia Amazon EC2, que puede alcanzar hasta 100 GB/s en una sola instancia. A continuación, estas aplicaciones agregan rendimiento en varias instancias para obtener varios terabits por segundo.

Otras aplicaciones son sensibles a la latencia, como las aplicaciones de mensajería de las redes sociales. Estas aplicaciones pueden lograr latencias para objetos pequeños coherentes (y latencias de “first-byte-out” para objetos más grandes) de unos 100-200 milisegundos aproximadamente.

Otros servicios de AWS también pueden ayudar a acelerar el rendimiento para diferentes arquitecturas de aplicaciones. Por ejemplo, si desea velocidades de transferencia mayores a través de una conexión HTTP única o latencias de milisegundos de un solo dígito, use Amazon CloudFront o Amazon ElastiCache para el almacenamiento en caché con Amazon S3.

De forma adicional, si desea transportar rápidamente los datos a largas distancias entre un cliente y un bucket de S3, use Configuración de transferencias de archivos rápidas y seguras con Amazon S3

Transfer Acceleration. Transfer Acceleration usa las ubicaciones de borde distribuidas globalmente en CloudFront para acelerar el transporte de los datos a través de grandes distancias geográficas. Si la carga de trabajo de Amazon S3 utiliza el cifrado del lado del servidor con AWS KMS, consulte Límites de AWS KMS en la Guía para desarrolladores de AWS Key Management Service para obtener información acerca de las tasas de solicitud admitidas para su caso de uso.

En los siguientes temas se describen las directrices y patrones de diseño recomendados a fin de optimizar el rendimiento para las aplicaciones que usan Amazon S3. Consulte Directrices de rendimiento de Amazon S3 y Patrones de diseño de rendimiento para Amazon S3 para obtener la información más reciente sobre la optimización del rendimiento de Amazon S3.

> **Note**
>
> Para obtener más información sobre el uso de la clase de almacenamiento Amazon S3 Express One Zone con buckets de directorio, consulte ¿Qué es S3 Express One Zone? y Buckets de directorio.

Temas

Directrices de rendimiento de Amazon S3

Al crear aplicaciones que cargan y recuperan objetos de Amazon S3, siga las directrices de nuestras prácticas recomendadas para optimizar el rendimiento. También ofrecemos con más detall Patrones de diseño de rendimiento.

Para obtener el mejor rendimiento para su aplicación en Amazon S3, recomendamos las siguientes directrices.

Temas

- Uso de recuperaciones de rango de byte
- Reintento de solicitudes de aplicaciones sensibles a la latencia
- Combinación de Amazon S3 (almacenamiento) y Amazon EC2 (cómputo) en la misma Región de AWS
- Uso de Amazon S3 Transfer Acceleration para minimizar la latencia generada por la distancia
- Uso de la versión más reciente de los SDK de AWS

Medición del rendimiento

Al optimizar el rendimiento, fíjese en los requisitos de rendimiento de red, CPU y DRAM. Dependiendo de la combinación de demandas de estos otros recursos, es posible que merezca la pena evaluar otros tipos de instancias Amazon EC2. Para obtener más información acerca de los tipos de instancias, consulte Instance types en la Guía del usuario de Amazon EC2.

También es útil fijarse en el tiempo de búsqueda de DNS, la latencia y la velocidad de transferencia de datos mediante herramientas de análisis HTTP al medir el rendimiento.

Para comprender los requisitos de rendimiento y optimizar el rendimiento de la aplicación, también puede supervisar las respuestas de error 503 que recibe. La monitorización de ciertas métricas de rendimiento puede generar gastos adicionales. Para obtener más información, consulte Precios de Amazon S3.

Monitorizar el número de respuestas de error de estado 503 (ralentización)

Para monitorizar el número de respuestas de error de estado 503 que recibe, puede utilizar una de las siguientes opciones:

- Use las métricas de solicitud de Amazon CloudWatch para Amazon S3. Las métricas de solicitud de CloudWatch incluyen una métrica para las respuestas de estado 5xx. Para obtener más información acerca de las métricas de solicitud de CloudWatch, consulte Monitorización de métricas con Amazon CloudWatch.
- Utilice el recuento de errores 503 (servicio no disponible) disponible en la sección de métricas avanzadas de Lente de almacenamiento de Amazon S3. Para obtener más información, consulte Uso de métricas de S3 Storage Lens para mejorar el rendimiento.
- Use el registro de acceso al servidor de Amazon S3. Con el registro de acceso al servidor, puede filtrar y revisar todas las solicitudes que reciben respuestas 503 (error interno). También puede

usar Amazon Athena para analizar los registros. Para obtener más información sobre el registro de acceso del servidor, consulte Registro de solicitudes con registro de acceso al servidor.

Al monitorizar la cantidad de códigos de error de estado HTTP 503, puede obtener información valiosa sobre qué prefijos, claves o buckets reciben la mayor cantidad de solicitudes de limitación.

Escalado horizontal de las conexiones de almacenamiento

La distribución de las solicitudes entre muchas conexiones es un patrón de diseño habitual para escalar horizontalmente el rendimiento. Al crear aplicaciones de alto rendimiento, piense en Amazon S3 como un sistema distribuido muy grande, no como un punto de enlace de una sola red como un servidor de almacenamiento tradicional. Puede lograr el mejor rendimiento emitiendo varias solicitudes simultáneas a Amazon S3. Distribuya estas solicitudes a través de conexiones separadas para maximizar el ancho de banda accesible desde Amazon S3. Amazon S3 no tiene límites en cuanto al número de conexiones que se realizan en su bucket.

Uso de recuperaciones de rango de byte

Al usar el encabezado HTTP `Range` en una solicitud GET Object, puede recuperar un rango de byte de un objeto, transfiriendo solo la parte especificada. Puede usar conexiones simultáneas a Amazon S3 para recuperar otros rangos de byte desde dentro del mismo objeto. Esto le ayuda a lograr un rendimiento total mayor frente a una sola solicitud de todo el objeto. La recuperación de rangos más pequeños de un objeto grande también permite que su aplicación mejore los tiempos de reintento al interrumpirse las solicitudes. Para obtener más información, consulte Descarga de objetos.

Los tamaños típicos para las solicitudes de rango de byte son 8 MB o 16 MB. Si los objetos aplican PUT mediante una carga multiparte, aplicarles GET en los mismos tamaños de parte (o al menos alinearlos con los límites de parte) es una buena práctica para lograr el mejor rendimiento. Las solicitudes GET pueden ocuparse directamente de partes individuales; por ejemplo, `GET ? partNumber=N.`

Reintento de solicitudes de aplicaciones sensibles a la latencia

Los reintentos y tiempos de espera agresivos contribuyen a potenciar una latencia coherente. Teniendo en cuenta la gran escala de Amazon S3, si la primera solicitud es lenta, es probable que una solicitud que se ha intentado de nuevo tome otra ruta y se realice correctamente. Los SDK de AWS cuentan con un tiempo de espera configurable y valores de reintento que puede ajustar a las tolerancias de su aplicación específica.

Combinación de Amazon S3 (almacenamiento) y Amazon EC2 (cómputo) en la misma Región de AWS

Aunque los nombres del bucket de S3 son únicos a nivel global, cada uno de ellos se almacena en una región que se selecciona al crearlos. Para optimizar el rendimiento, recomendamos que acceda al bucket desde las instancias de Amazon EC2 en la misma Región de AWS cuando sea posible. Esto ayuda a reducir los costos de la transferencia de datos y la latencia de red.

Para obtener más información acerca de los costes de las transferencias de datos, consulte Precios de Amazon S3.

Uso de Amazon S3 Transfer Acceleration para minimizar la latencia generada por la distancia

Configuración de transferencias de archivos rápidas y seguras con Amazon S3 Transfer Acceleration administra transferencias de archivos rápidas, fáciles y seguras a través de grandes distancias geográficas entre el cliente y un bucket de S3. Transfer Acceleration aprovecha las ubicaciones de borde distribuidas globalmente en Amazon CloudFront. A medida que los datos llegan a una ubicación de borde, se redirigen a Amazon S3 a través de una ruta de red optimizada. Transfer Acceleration es ideal para transferir desde gigabytes hasta terabytes de datos con frecuencia entre continentes. También resulta útil para los clientes que cargan en un bucket centralizado desde todo el mundo.

Puede utilizar la herramienta de comparación de velocidad de Amazon S3 Transfer Acceleration para comparar velocidades de subida aceleradas y no aceleradas en las regiones de Amazon S3. La herramienta de comparación de velocidad utiliza cargas multipartes para transferir un archivo desde su navegador hacia diversas regiones de Amazon S3 con y sin Amazon S3 Transfer Acceleration.

Uso de la versión más reciente de los SDK de AWS

Los SDK de AWS ofrecen compatibilidad integrada con muchas de las directrices recomendadas para optimizar el rendimiento de Amazon S3. Asimismo, proporcionan una API más sencilla para aprovechar Amazon S3 desde dentro de una aplicación y se actualizan con frecuencia para seguir las prácticas recomendadas más recientes. Por ejemplo, los SDK incluyen lógica para reintentar solicitudes automáticamente en errores HTTP 503 e invierten en código para responder a las conexiones lentas y adaptarse a ellas.

Los SDK también ofrecen un gestor de transferencias, que automatiza el escalado horizontal de conexiones para lograr miles de solicitudes por segundo, empleando solicitudes de rango de

byte si procede. Es importante usar la versión más reciente de los SDK de AWS para obtener las características de optimización de rendimiento más recientes.

También puede optimizar el rendimiento al usar solicitudes de la API de REST de HTTP. Al usar la API de REST, debe seguir las mismas prácticas recomendadas que forman parte de los SDK. Permita los tiempos de espera y los reintentos en las solicitudes lentas y varias conexiones para que la recuperación de datos de objeto en paralelo sea posible. Para obtener información sobre el uso de la API de REST, consulte la referencia de la API de Amazon Simple Storage Service.

Patrones de diseño de rendimiento para Amazon S3

Al diseñar aplicaciones para cargar y recuperar objetos de Amazon S3, use los patrones de diseño de nuestras prácticas recomendadas para lograr el mejor rendimiento para su aplicación. También ofrecemos Directrices de rendimiento para que las tenga en cuenta al planificar la arquitectura de aplicaciones.

Para optimizar el rendimiento, puede usar los siguientes patrones de diseño.

Temas

Uso del almacenamiento en caché para el contenido de acceso frecuente

Muchas aplicaciones que almacenan datos en Amazon S3 ofrecen un "conjunto de trabajo" que los usuarios solicitan continuamente. Si una carga de trabajo envía solicitudes GET repetidas para un conjunto común de objetos, puede utilizar una caché como Amazon CloudFront, Amazon ElastiCache o AWS Elemental MediaStore para optimizar el rendimiento. La adopción correcta de la caché puede dar lugar a una baja latencia y a velocidades de transferencias de datos altas. Las aplicaciones que usan el almacenamiento en caché también envían menos solicitudes directas a Amazon S3, lo que puede contribuir a reducir los costes de las solicitudes.

Amazon CloudFront es una red de entrega de contenido (CDN) rápida que almacena datos en caché de forma transparente desde Amazon S3 en un gran conjunto de puntos de presencia (PoP)

distribuidos geográficamente. Cuando se puede tener acceso a los objetos desde multirregiones o a través de Internet, CloudFront permite que los datos se almacenen en caché cerca de los usuarios con acceso a los objetos. Esto puede dar como resultado la entrega de alto rendimiento de contenido popular de Amazon S3. Para obtener más información sobre CloudFront, consulte la guía para desarrolladores de Amazon CloudFront.

Amazon ElastiCache es una caché en memoria administrada. Con ElastiCache, puede aprovisionar instancias Amazon EC2 que almacenan en caché objetos en memoria. Este almacenamiento en caché se traduce en pedidos de reducción de la magnitud en la latencia GET y aumentos sustanciales en el rendimiento de descarga. Para usar ElastiCache, debe modificar la lógica de la aplicación tanto para rellenar la caché con objetos activos como para comprobar la caché en busca de estos objetos antes de solicitarlos en Amazon S3. Para ver ejemplos de uso de ElastiCache para mejorar el rendimiento de GET de Amazon S3, consulte la publicación del blog Turbocharge Amazon S3 with Amazon ElastiCache for Redis.

AWS Elemental MediaStore es un sistema de almacenamiento en caché y distribución de contenido creado específicamente para flujos de trabajo de vídeo y entrega de medios desde Amazon S3. MediaStore proporciona API de almacenamiento integrales específicamente para vídeo, y está recomendado para cargas de trabajo de video sensibles al rendimiento. Para obtener información sobre MediaStore, consulte la Guía del usuario de AWS Elemental MediaStore.

Tiempos de espera y reintentos de aplicaciones sensibles a la latencia

Hay ciertas situaciones en las que una aplicación recibe una respuesta de Amazon S3 que indica que es necesario volver a intentarlo. Amazon S3 asigna nombres de bucket y objeto a los datos de objeto asociados a ellos. Si una aplicación genera velocidades de solicitudes altas (normalmente velocidades sostenidas de más de 5000 solicitudes por segundo a un pequeño número de objetos), podría recibir respuestas de ralentización HTTP 503. Si se producen estos errores, cada SDK de AWS implementa la lógica de reintentos automática mediante el retroceso exponencial. Si no está usando un SDK de AWS, debe implementar la lógica de reintentos al recibir el error HTTP 503. Para obtener información acerca de las técnicas de retardo, consulte Reintentos de error y retroceso exponencial en AWS en la Referencia general de Amazon Web Services.

Amazon S3 se escala automáticamente en respuesta a las nuevas velocidades de solicitudes sostenidas, optimizando el rendimiento de forma dinámica. Aunque Amazon S3 se está optimizando internamente para una nueva velocidad de solicitudes, recibirá respuestas a las solicitudes HTTP 503 de forma temporal hasta que se complete la optimización. Una vez que Amazon S3 optimice

internamente el rendimiento para la nueva velocidad de las solicitudes, todas las solicitudes se atienden de forma general sin reintentos.

En las aplicaciones sensibles a la latencia, Amazon S3 aconseja un seguimiento y realizar un reintento agresivo de operaciones más lentas. Siempre que reintente una solicitud, recomendamos que se use una nueva conexión a Amazon S3 y que se vuelva a realizar una búsqueda de DNS.

Si realiza solicitudes de tamaño grande y variable (por ejemplo: más de 128 MB), aconsejamos que se realice un seguimiento del rendimiento logrado y que se reintente el 5 % más lento de las solicitudes. Al realizar solicitudes más pequeñas (por ejemplo: menos de 512 KB), donde las latencias medias suelen situarse en el rango de las decenas de milisegundos, una buena directriz es reintentar una operación GET o PUT transcurridos 2 segundos. Si son necesarios reintentos adicionales, la práctica recomendada es el retardo. Por ejemplo, recomendamos que se emita un reintento transcurridos 2 segundos y un segundo reintento después de 4 segundos adicionales.

Si su aplicación realiza solicitudes de tamaño fijo a Amazon S3, debe esperar unos tiempos de respuesta más uniformes para cada una de estas solicitudes. En este caso, una estrategia sencilla consiste en identificar el 1 % más lento de las solicitudes y reintentarlas. Incluso un único reintento suele ser eficaz reduciendo la latencia.

Si está utilizando AWS Key Management Service (AWS KMS) para el cifrado del lado del servidor, consulte Límites en la Guía para desarrolladores de AWS Key Management Service con el fin de obtener información acerca de las tasas de solicitudes admitidas para su caso de uso.

Escalado horizontal y uso en paralelo de solicitudes para lograr un alto rendimiento

Amazon S3 es un sistema distribuido muy grande. Para ayudarle a aprovechar su escala, le animamos a escalar horizontalmente solicitudes paralelas a los puntos de enlace de servicio de Amazon S3. Además de distribuir las solicitudes en Amazon S3, este tipo de enfoque de escalado ayuda a distribuir la carga mediante varias rutas a través de la red.

Para las transferencias de alto rendimiento, Amazon S3 aconseja que se usen aplicaciones que a su vez usen varias conexiones a los datos de GET o PUT en paralelo. Por ejemplo, esto cuenta con el respaldo del gestor de transferencias de Amazon S3 en el SDK de AWS para Java. Además, la mayoría de los otros SDK de AWS proporcionan construcciones similares. Para algunas aplicaciones, puede lograr conexiones paralelas lanzando varias solicitudes simultáneamente en diferentes subprocesos de aplicación, o bien en diferentes instancias de aplicación. El mejor enfoque que adoptar depende de su aplicación y la estructura de los objetos a los que tiene acceso.

Puede utilizar los SDK de AWS para emitir las solicitudes GET y PUT directamente en lugar de emplear la administración de las transferencias en el SDK de AWS. Este enfoque le permite ajustar su carga de trabajo de forma más directa, mientras sigue beneficiándose de la compatibilidad del SDK con los reintentos y su control de cualquier respuesta HTTP 503 que pueda surgir. Como regla general, al descargar objetos grandes dentro de una región desde Amazon S3 a Amazon EC2, recomendamos que se realicen solicitudes simultáneas de rangos de byte de un objeto en la granularidad de 8-16 MB. Realice una solicitud simultánea de cada valor comprendido en un intervalo de 85-90 MB/s del rendimiento de red deseado. Para saturar una tarjeta de interfaz de red (NIC), puede usar unas 15 solicitudes simultáneas a través de conexiones independientes. Puede escalar de forma ascendente las solicitudes simultáneas a través de más conexiones para saturar las NIC con mayor rapidez, como NIC de 25 GB/s y de 100 GB/s.

Medir el rendimiento es importante cuando ajusta el número de solicitudes que se van a emitir simultáneamente. Recomendamos comenzar con una sola solicitud cada vez. Mida el ancho de banda de red logrado y el uso de otros recursos utilizados por su aplicación durante el procesamiento de los datos. A partir de ese momento, podrá identificar el recurso de cuello de botella (es decir, el recurso más usado) y, por tanto, el número de solicitudes con probabilidades de resultar de utilidad. Por ejemplo, si el procesamiento de una solicitud cada vez se traduce a un uso del 25 % de la CPU, sugiere que se pueden atender hasta cuatro solicitudes simultáneas. La medición es fundamental y merece la pena confirmar el uso de recursos a medida que aumenta la velocidad de solicitudes.

Si su aplicación emite solicitudes directamente a Amazon S3 mediante la API de REST, recomendamos usar un grupo de conexiones HTTP y volver a utilizar cada conexión para una serie de solicitudes. Al evitarse la configuración de la conexión por solicitud, desaparece la necesidad de llevar a cabo protocolos de enlace de Capa de conexión segura (SSL) y TCP de inicio lento. Para obtener información sobre el uso de la API de REST, consulte la referencia de la API de Amazon Simple Storage Service.

Por último, merece la pena prestar atención a DNS y volver a comprobar si las solicitudes se distribuyen mediante un amplio grupo de direcciones IP de Amazon S3. Consultas de DNS para el ciclo de Amazon S3 a través de una gran lista de puntos de enlace de IP. Sin embargo, el almacenamiento en caché de los solucionadores o el código de aplicación que vuelve a usar una sola dirección IP no se beneficia de la diversidad de direcciones y el balanceo de carga que se produce a continuación. Las herramientas de utilidades de red, como, por ejemplo, la herramienta de línea de comandos `netstat`, pueden mostrar las direcciones IP que se están utilizando para la comunicación con Amazon S3. Además, proporcionamos directrices acerca de las configuraciones

de DNS que se deben emplear. Para obtener más información acerca de estas pautas, consulte Realizar solicitudes.

Uso de Amazon S3 Transfer Acceleration para acelerar las transferencias de datos a lugares geográficos dispares

Configuración de transferencias de archivos rápidas y seguras con Amazon S3 Transfer Acceleration resulta eficaz a la hora de minimizar o eliminar la latencia generada por la distancia geográfica existente entre clientes repartidos por todo el mundo y una aplicación regional mediante Amazon S3. Transfer Acceleration usa las ubicaciones de borde distribuidas globalmente en CloudFront para el transporte de los datos. La red de borde de AWS tiene puntos de presencia en más de 50 ubicaciones. Actualmente, se usa para distribuir el contenido a través de CloudFront y proporcionar respuestas rápidas a las consultas DNS realizadas a Amazon Route 53.

La red de borde también ayuda a acelerar las transferencias de datos tanto dentro como fuera de Amazon S3. Resulta ideal para las aplicaciones que transfieren datos en o entre continentes, tienen una conexión a Internet rápida, usan objetos grandes o tienen mucho contenido que cargar. A medida que los datos llegan a una ubicación de borde, se redirigen a Amazon S3 a través de una ruta de red optimizada. En general, cuanto más lejos esté de una región de Amazon S3, mayor será la mejora de la velocidad que puede esperar del uso de Transfer Acceleration.

Puede configurar Transfer Acceleration en buckets nuevos o ya existentes. Puede usar un punto de enlace independiente de Amazon S3 Transfer Acceleration para utilizar las ubicaciones de borde de AWS. La mejor forma de probar si Transfer Acceleration contribuye al rendimiento de las solicitudes de los clientes es usar la herramienta de comparación de velocidad de Amazon S3 Transfer Acceleration. Las condiciones y configuraciones de red varían de cuando en cuando y de ubicación a ubicación. Así pues, solo se le cobrarán las transferencias en las que Amazon S3 Transfer Acceleration pueda mejorar de forma potencial su rendimiento de carga. Para obtener información acerca del uso de Transfer Acceleration con diferentes SDK de AWS, consulte Habilitación y uso de S3 Transfer Acceleration.

¿Qué es Amazon S3 en Outposts?

AWS Outposts es un servicio totalmente administrado que ofrece la misma infraestructura de AWS, servicios de AWS, API y herramientas de prácticamente cualquier centro de datos, espacio de coubicación o instalación local para obtener una experiencia híbrida realmente uniforme.AWS Outposts es ideal para cargas de trabajo que requieren acceso de baja latencia a sistemas en las instalaciones, procesamiento de datos local, residencia de datos y migración de aplicaciones con interdependencias de sistemas locales. Para obtener más información, consulte ¿Qué es AWS Outposts? en la Guía del usuario de AWS Outposts.

Con Amazon S3 en Outposts, puede crear buckets de S3 en sus Outposts y almacenar y recuperar objetos fácilmente en las instalaciones. S3 en Outposts proporciona una nueva clase de almacenamiento, `OUTPOSTS`, que utiliza las API de Amazon S3 y está diseñada para almacenar datos de manera duradera y redundante en múltiples dispositivos y servidores de Outposts. Usted se comunica con su bucket de Outposts mediante un punto de acceso y una conexión de punto de conexión a través de una nube privada virtual (VPC).

Puede usar las mismas API y características en los buckets de Outposts que en Amazon S3, como políticas de acceso, cifrado y etiquetado. Puede utilizar S3 en Outposts a través de la AWS Management Console, AWS Command Line Interface (AWS CLI), AWS SDK o la API de REST.

Cómo funciona S3 en Outposts

S3 en Outposts es un servicio de almacenamiento de objetos que almacena datos como objetos dentro de buckets en Outpost. Un objeto es un archivo de datos y cualquier metadato que describa ese archivo. Un bucket es un contenedor de objetos.

Para almacenar datos en S3 en Outposts, primero debe crear un bucket. Al crear el bucket, debe especificar un nombre de bucket y el Outpost que lo contendrá. Para acceder a su bucket de S3 en

Outposts y realizar operaciones en objetos, debe crear y configurar un punto de acceso. También debe crear un punto de conexión para enrutar las solicitudes al punto de acceso.

Los puntos de acceso facilitan el acceso a datos para cualquier Servicio de AWS o aplicación de cliente que almacena datos en S3. Los puntos de acceso son puntos de conexión de red con nombre asociados a los buckets que se pueden utilizar para realizar operaciones con objetos, como `GetObject` y `PutObject`. Cada punto de acceso tiene permisos y controles de red distintos.

Puede crear y administrar sus buckets de S3 en Outposts, puntos de acceso y puntos de conexión mediante la AWS Management Console, AWS CLI, SDK de AWS o API de REST. Para cargar y administrar objetos en su bucket de S3 en Outposts, puede utilizar la AWS CLI, SDK de AWS o API de REST.

Regiones

Durante el aprovisionamiento de AWS Outposts, usted o AWS crea una conexión de enlace de servicio que conecta su Outpost de nuevo a la Región de AWS elegida o la región de origen de Outposts para operaciones de buckets y telemetría. Un Outpost depende de la conectividad con la Región de AWS principal. El bastidor de Outposts no está diseñado para operaciones o entornos desconectados con conectividad limitada o nula. Para obtener más información, consulte Conectividad de Outpost a Regiones de AWS en la Guía del usuario de AWS Outposts.

Buckets

Un bucket es un contenedor para objetos almacenados en S3 en Outposts. Puede almacenar cualquier cantidad de objetos en un bucket y puede tener hasta 100 buckets por cuenta y Outpost.

Cuando cree un bucket, introduzca un nombre de bucket y elija el Outpost donde residirá el bucket. Después de crear un bucket, no se puede cambiar su nombre ni moverlo a otro Outpost. Los nombres de los buckets deben cumplir las Reglas de nomenclatura de buckets de Amazon S3. En S3 en Outposts, los nombres de buckets son únicos de un Outpost y Cuenta de AWS. Los buckets de S3 en Outposts requieren el `outpost-id`, el `account-id` y el nombre del bucket para identificarlos.

En el siguiente ejemplo, se muestra el formato de nombre de recurso de Amazon (ARN) para los buckets de S3 en Outposts. El ARN consta de la región a la que está destinado el Outpost, su cuenta de Outpost, el ID de Outpost y el nombre del bucket.

```
arn:aws:s3-outposts:region:account-id:outpost/outpost-id/bucket/bucket-name
```

Cada objeto está almacenado en un bucket. Debe utilizar puntos de acceso para acceder a cualquier objeto de un bucket de Outposts. Cuando especifica el bucket para las operaciones de objetos, se utiliza el ARN del punto de acceso o el alias del punto de acceso. Para obtener más información acerca de los alias de punto de acceso, consulte Uso de un alias de estilo de bucket para su punto de acceso de bucket de S3 en Outposts.

En el siguiente ejemplo se muestra el formato de ARN del punto de acceso para S3 en Outposts, que incluye el `outpost-id`, `account-id`, y el nombre del punto de acceso:

```
arn:aws:s3-outposts:region:account-id:outpost/outpost-id/accesspoint/accesspoint-name
```

Para obtener más información acerca de los buckets, consulte Trabajo con buckets de S3 en Outposts.

Objects

Los objetos son las entidades fundamentales almacenadas en S3 en Outposts. Los objetos se componen de datos de objetos y metadatos. Los metadatos son conjuntos de pares nombre-valor que describen el objeto. Incluyen algunos metadatos predeterminados, como la fecha de la última modificación y los metadatos HTTP estándar, como `Content-Type`. También puede especificar metadatos personalizados en el momento en que se almacena el objeto. Un objeto se identifica de forma exclusiva dentro de un bucket con una clave (o nombre).

Con Amazon S3 en Outposts, los datos de objeto siempre se almacenan en el Outpost. Cuando AWS instala un bastidor de Outpost, sus datos permanecen de manera local en su Outpost para cumplir los requisitos de residencia de datos. Sus objetos nunca salen de su Outpost y no están en una Región de AWS. Ya que la AWS Management Console está alojada dentro de la región, no puede usar la consola para cargar o administrar objetos en su Outpost. Sin embargo, puede utilizar la API de REST, AWS Command Line Interface (AWS CLI) y los SDK de AWS para cargar y administrar los objetos a través de los puntos de acceso.

Claves

Una clave de objeto (o nombre de clave) es el identificador único de un objeto dentro de un bucket. Cada objeto de un bucket tiene exactamente una clave. La combinación de un bucket y clave de objeto identifica de forma única cada objeto.

En el siguiente ejemplo, se muestra el formato ARN para los objetos de S3 en Outposts, que incluye el código de Región de AWS para la región a la que está destinado el Outpost, el ID de Cuenta de AWS, el ID de Outpost, el nombre del bucket y la clave de objeto:

```
arn:aws:s3-outposts:us-west-2:123456789012:outpost/ op-01ac5d28a6a232904/bucket/amzn-s3-demo-bucket1/object/myobject
```

Para obtener más información sobre las claves de objetos, consulte Trabajo con objetos de S3 en Outposts.

Control de versiones de S3

Puede usar el control de versiones de S3 en buckets de Outposts para conservar diversas variantes de un objeto en el mismo bucket. Puede utilizar el control de versiones de S3 para conservar, recuperar y restaurar todas las versiones de los objetos almacenados en su bucket de . EL control de versiones de S3 ayuda a recuperarse de acciones no deseadas del usuario y de errores de la aplicación.

Para obtener más información, consulte Administración de control de versiones de S3 para su bucket de S3 en Outposts.

ID de versión.

Si activa el control de versiones de S3 en un bucket, S3 en Outposts genera un ID de versión único para cada objeto agregado al bucket. Los objetos que ya existían en el bucket en el momento en que habilita el control de versiones tienen un ID de versión de `null`. Si modifica estos objetos (o cualquier otro) con otras operaciones, como PutObject, los objetos nuevos obtienen un ID de versión único.

Para obtener más información, consulte Administración de control de versiones de S3 para su bucket de S3 en Outposts.

Clase de almacenamiento y cifrado

S3 en Outposts proporciona una nueva clase de almacenamiento, S3 Outposts (`OUTPOSTS`). La clase de almacenamiento de S3 Outposts solo está disponible para los objetos almacenados en buckets que se encuentran en AWS Outposts. Si intenta usar otras clases de almacenamiento de S3 con S3 en Outposts, S3 en Outposts devuelve el error `InvalidStorageClass`.

De manera predeterminada, los objetos almacenados en la clase de almacenamiento S3 Outposts (OUTPOSTS) siempre se cifran mediante cifrado del lado del servidor con claves de cifrado administradas (SSE-S3) de Amazon S3. Para obtener más información, consulte Cifrado de datos en S3 en Outposts.

Política de bucket

Una política de bucket es una política AWS Identity and Access Management basada en recursos (IAM) que puede utilizar para conceder permisos de acceso al bucket y a los objetos que contiene. Solo el propietario del bucket puede asociar una política a un bucket. Los permisos asociados a un bucket se aplican a todos los objetos del bucket que son propiedad de la cuenta de propietario del bucket. Las políticas de bucket tienen un límite de tamaño de 20 KB.

Las políticas de buckets utilizan el lenguaje de políticas de IAM basado en JSON que es estándar en AWS. Puede utilizar directivas de bucket para agregar o denegar permisos para los objetos de un bucket. Las políticas de bucket permiten o deniegan solicitudes en función de los elementos de la política. Estos elementos puedne incluir el solicitante, las acciones de S3 en Outposts, los recursos y los aspectos o condiciones de la solicitud (por ejemplo: la dirección IP utilizada para realizar la solicitud). Por ejemplo, puede crear una política de bucket que otorgue permisos entre cuentas para cargar objetos en un bucket de S3 en Outpost y, al mismo tiempo, garantizar que el propietario del bucket tenga el control total de los objetos cargados. Para obtener más información, consulte Ejemplos de políticas de bucket de Amazon S3.

En su política de bucket, puede utilizar caracteres comodín (*) en ARN y otros valores para otorgar permisos a un subconjunto de objetos. Por ejemplo, puede controlar el acceso a grupos de objetos que empiezan por un prefijo o terminar con una extensión dada, como `.html`.

Puntos de acceso de S3 en Outposts

Los puntos de acceso de S3 en Outpost se denominan puntos de conexión de red con políticas de acceso dedicadas que describen cómo se puede acceder a los datos mediante ese punto de conexión. Los puntos de acceso simplifican la administración del acceso a los datos a escala para los conjuntos de datos compartidos en S3 en Outposts. Los puntos de acceso se asocian a los buckets que se pueden utilizar para realizar operaciones con objetos de S3, como `GetObject` y `PutObject`.

Cuando especifica el bucket para las operaciones de objetos, se utiliza el ARN del punto de acceso o el alias del punto de acceso. Para obtener más información acerca de los alias de punto de acceso, consulte Uso de un alias de estilo de bucket para su punto de acceso de bucket de S3 en Outposts.

Cada punto de acceso tiene permisos y controles de red distintos que S3 en Outposts se aplica a cualquier solicitud que se realice a través de ese punto de acceso. Cada punto de acceso aplica una política de punto de acceso personalizada que funciona en conjunción con la política de bucket asociada al bucket subyacente.

Para obtener más información, consulte Acceso a los buckets y objetos de S3 en Outposts.

Características de S3 en Outposts

Administración de accesos

Amazon S3 proporciona características para auditar y administrar el acceso a sus buckets y objetos. De forma predeterminada, los buckets de S3 en Outposts y los objetos que hay en ellos son privados. Solo tiene acceso a los recursos de S3 en Outpost que cree.

Para conceder permisos de recursos detallados que admitan su caso de uso específico o para auditar los permisos de sus recursos de S3 en Outpost, puede utilizar las siguientes características.

- Bloqueo del acceso público de S3: bloquea el acceso público a los buckets y objetos. Para buckets en Outposts, el bloqueo del acceso público siempre está habilitado de forma predeterminada.
- AWS Identity and Access Management (IAM): IAM es un servicio web que le ayuda a controlar de forma segura el acceso a los recursos de AWS, como los recursos de S3 en Outposts. Con IAM, se pueden administrar de forma centralizada los permisos que controlan a qué recursos de AWS pueden acceder los usuarios. Utilice IAM para controlar quién está autenticado (ha iniciado sesión) y autorizado (tiene permisos) para utilizar recursos.
- Puntos de acceso de S3 en Outposts: administre el acceso a los datos para los conjuntos de datos compartidos en S3 en Outposts. Los puntos de acceso se denominan puntos de conexión de red con nombre con políticas de acceso dedicadas. Los puntos de acceso se asocian a los buckets y se pueden utilizar para realizar operaciones con objetos, como `GetObject` y `PutObject`.
- Políticas de buckets: utilice el lenguaje de políticas basado en IAM para configurar permisos basados en recursos para los buckets de S3 y los objetos que hay en ellos.
- AWS Resource Access Manager (AWS RAM): comparta de forma segura su capacidad de S3 en Outposts en Cuentas de AWS, dentro de su organización o en unidades organizativas (OU) en AWS Organizations.

Registro y monitorización

S3 en Outposts proporciona herramientas de registro y supervisión que puede utilizar para supervisar y controlar cómo se utilizan sus recursos de S3 en Outposts. Para obtener más información sobre la monitorización de , consulte .

- Métricas de Amazon CloudWatch para S3 en Outposts: realice un seguimiento del estado operativo de sus recursos y conozca la disponibilidad de su capacidad.
- Eventos de Amazon CloudWatch Events para S3 en Outposts: cree una regla para cualquier evento de API de S3 en Outposts para recibir notificaciones en todos los destinos de CloudWatch Events compatibles, incluidos Amazon Simple Queue Service (Amazon SQS), Amazon Simple Notification Service (Amazon SNS) y AWS Lambda.
- Registros de AWS CloudTrail para S3 en Outposts: registre las medidas adoptadas por un usuario, un rol o un Servicio de AWS en S3 en Outposts. Los registros de CloudTrail le proporcionan un seguimiento detallado de la API para las operaciones a nivel de bucket y de objeto de Amazon S3.

Consistencia sólida

S3 en Outposts proporciona una sólida coherencia de lectura tras escritura para las solicitudes PUT y DELETE de objetos del bucket de S3 en Outposts en todas las Regiones de AWS. Este comportamiento se aplica tanto a las escrituras en objetos nuevos como a las solicitudes PUT que sobrescriben objetos existentes y las solicitudes DELETE. Además, las etiquetas de objeto y los metadatos de objetos de S3 en Outposts (p. ej., el objeto HEAD) tienen una buena coherencia. Para obtener más información, consulte Modelo de consistencia de datos de Amazon S3.

Servicios relacionados

Una vez que cargue sus datos en S3 en Outposts, puede utilizarlos con otros Servicios de AWS. Los siguientes servicios son los que puede utilizar con más frecuencia:

- Amazon Elastic Compute Cloud (Amazon EC2): proporciona capacidad de computación escalable y segura en Nube de AWS. El uso de Amazon EC2 reduce la necesidad de invertir inicialmente en hardware, de manera que puede desarrollar e implementar aplicaciones en menos tiempo. Puede usar Amazon EC2 para lanzar tantos servidores virtuales como necesite, configurar la seguridad y las redes, y administrar el almacenamiento.

- Amazon Elastic Block Store (Amazon EBS) on Outposts: utilice Amazon EBS en instantáneas locales en Outposts para almacenar instantáneas de volúmenes en un Outpost localmente en S3 en Outposts.
- Amazon Relational Database Service (Amazon RDS) en Outposts: utilice las copias de seguridad locales de Amazon RDS para almacenar sus copias de seguridad de Amazon RDS localmente en su Outpost.
- AWS DataSync: automatice la transferencia de datos entre Outposts y Regiones de AWS mediante la elección de lo que se va a transferir, cuándo se va a transferir y cuánto ancho de banda de red se va a usar. S3 en Outposts se integra con AWS DataSync. Para las aplicaciones en las instalaciones que requieren un procesamiento local de alto rendimiento, S3 en Outposts proporciona almacenamiento de objetos en las instalaciones con el fin de minimizar las transferencias de datos y el búfer de las variaciones de red, al tiempo que permite transferir datos con facilidad entre Outposts y las Regiones de AWS.

Acceso a S3 en Outposts

Puede trabajar con S3 en Outposts de cualquiera de las siguientes formas:

AWS Management Console

La consola es una interfaz de usuario basada en la web para administrar S3 en Outposts y los recursos de AWS. Si se ha registrado en una Cuenta de AWS, puede acceder a S3 en Outposts iniciando sesión en la AWS Management Console y eligiendo S3 en la página de inicio de AWS Management Console. A continuación, elija Outposts buckets (Buckets de Outposts) desde el panel de navegación izquierdo.

AWS Command Line Interface

Puede utilizar las herramientas de línea de comandos de AWS para emitir comandos o compilar scripts en la línea de comandos de su sistema con el fin de ejecutar tareas de AWS (incluidas las de S3).

AWS Command Line Interface (AWS CLI) proporciona comandos para una amplia gama de Servicios de AWS. La AWS CLI es compatible con Windows, macOS y Linux. Para empezar, consulte la AWS Command Line Interface Guía de usuario de . Para obtener más información acerca de los comandos que puede usar con S3 en Outposts, consultes3api, s3control y s3outposts en la Referencia de comandos de AWS CLI.

SDK de AWS

AWS ofrece SDK (kits de desarrollo de software) que se componen de bibliotecas y código de muestra para diversos lenguajes de programación y plataformas (Java, Python, Ruby, .NET, iOS, Android, etc.). Los SDK de AWS proporcionan una forma cómoda de crear acceso de programación a S3 en Outposts y AWS. Dado que S3 en Outposts utiliza los mismos SDK que Amazon S3, S3 en Outposts proporciona una experiencia coherente utilizando las mismas API, automatización y herramientas de S3.

S3 en Outposts es un servicio REST. Puede enviar solicitudes a S3 en Outposts usando las bibliotecas de los SDK de AWS, que envuelven la API de REST subyacente y simplifican sus tareas de programación. Por ejemplo, los SDK se encargan de tareas como calcular firmas, firmar solicitudes criptográficamente, gestionar los errores y reintentar las solicitudes de forma automática. Para obtener información sobre los SDK de AWS (por ejemplo: cómo descargarlos e instalarlos), consulte Herramientas para crear en AWS.

Pago de S3 en Outposts

Puede comprar una amplia variedad de configuraciones de bastidor de AWS Outposts que incluyen una combinación de tipos de instancias de Amazon EC2, volúmenes de unidades de estado sólido (SSD) de uso general de Amazon EBS (gp2) y S3 en Outposts. Los precios incluyen entrega, instalación y mantenimiento de servicios de infraestructura y parches y actualizaciones de software.

Para obtener más información, consulte Precios de bastidores de AWS Outposts.

Siguientes pasos

Para obtener más información sobre cómo trabajar con S3 en Outposts, consulte los siguientes temas:

- Configuración de Outpost de
- ¿En qué se diferencia Amazon S3 en Outposts de Amazon S3?
- Introducción a Amazon S3 en Outposts
- Redes para S3 en Outposts
- Trabajo con buckets de S3 en Outposts
- Trabajo con objetos de S3 en Outposts

- Seguridad en S3 en Outposts
- Administración de almacenamiento de S3 en Outposts
- Desarrollo con Amazon S3 en Outposts

Configuración de Outpost de

Para comenzar a utilizar Amazon S3 en Outposts necesita una publicación de salida con capacidad de Amazon S3 implementada en sus instalaciones. Para obtener información acerca de las opciones para solicitar una capacidad de Outpost y S3, consulte AWS Outposts. Para comprobar si sus Outposts tienen capacidad para S3, puede usar la llamada a la API ListOutpostsWithS3. Para obtener más información sobre las especificaciones y ver en qué se diferencia S3 en Outposts de Amazon S3, consulte ¿En qué se diferencia Amazon S3 en Outposts de Amazon S3?

Para obtener más información, consulte los siguientes temas.

Temas

- Solicite un nuevo Outpost de

Solicite un nuevo Outpost de

Si necesita solicitar un nuevo Outpost con capacidad de S3, consulte los precios de AWS Outposts para comprender las opciones de capacidad de Amazon Elastic Compute Cloud (Amazon EC2), Amazon Elastic Block Store (Amazon EBS) y Amazon S3.

Después de seleccionar la configuración, siga los pasos que se muestran en Crear un Outpost y solicitar la capacidad de Outpost en la Guía del usuario de AWS Outposts.

¿En qué se diferencia Amazon S3 en Outposts de Amazon S3?

Amazon S3 en Outposts ofrece almacenamiento de objetos a su entorno de AWS Outposts en las instalaciones. S3 en Outposts le ayuda a satisfacer el procesamiento local, la residencia de datos y las exigentes necesidades de rendimiento al mantener los datos cerca de las aplicaciones en las instalaciones. Debido a que usa las API y funciones de Amazon S3, S3 en Outposts facilita el almacenamiento, la seguridad, el etiquetado, la elaboración de informes y el control del acceso a los datos de sus Outposts y amplía la infraestructura de AWS a su instalación local para obtener una experiencia híbrida uniforme.

Para obtener más información acerca de cómo S3 en Outposts es único, consulte los siguientes temas.

Temas

Especificaciones de S3 en Outposts

- El tamaño máximo del bucket de Outposts es de 50 TB.
- La cantidad máxima de buckets de Outposts es de 100 por Cuenta de AWS.
- Solo se puede acceder a los buckets de Outposts mediante puntos de acceso y puntos de conexión.
- El número máximo de puntos de acceso por bucket de Outposts es 10.
- Las políticas de punto de acceso tienen un límite de tamaño de 20 KB.
- El propietario de Outpost puede administrar el acceso dentro de la organización en AWS Organizations con AWS Resource Access Manager. Todas las cuentas que necesitan acceso a Outpost deben estar dentro de la misma organización que la cuenta de propietario en AWS Organizations.
- La cuenta de propietario del bucket S3 en Outposts siempre es propietaria de todos los objetos del bucket.
- Sólo la cuenta de propietario del bucket S3 en Outposts puede realizar operaciones en el bucket.
- Las limitaciones de tamaño de objeto son coherentes con Amazon S3.
- Todos los objetos almacenados en S3 en Outposts se almacenan en la clase de almacenamiento de `OUTPOSTS`.
- De forma predeterminada, todos los objetos almacenados en la clase de almacenamiento `OUTPOSTS` se almacenan mediante cifrado del lado del servidor con claves de cifrado administradas por Amazon S3 (SSE-S3). También puede elegir almacenar objetos mediante cifrado del lado del servidor con claves de cifrado proporcionadas por el cliente (SSE-C).
- Si no hay suficiente espacio para almacenar un objeto en su Outpost, la API devuelve una excepción de capacidad insuficiente (ICE).

Operaciones de la API compatibles con S3 en Outposts

Para ver la lista de operaciones de la API que se admiten en S3 en Outposts, consulte Operaciones de la API de Amazon S3 en Outposts.

Características de Amazon S3 no compatibles con S3 en Outposts

Las siguientes características de Amazon S3 no son actualmente compatibles con Amazon S3 en Outposts. Se rechaza cualquier intento de usarlas.

- Listas de control de acceso (ACL)
- Uso compartido de recursos entre orígenes (CORS)
- Operaciones por lotes de S3
- Informes de inventario de S3
- Cambio del cifrado predeterminado del bucket
- Buckets públicos
- Eliminación de la autenticación multifactor (MFA)
- Transiciones del ciclo de vida de S3 (además de la eliminación de objetos y la detención de cargas multiparte incompletas)
- Bloqueo de objetos de retención legal en S3
- Retención de bloqueo de objetos
- Cifrado del lado del servidor con claves AWS Key Management Service (AWS KMS) (SSE-KMS)
- Control del tiempo de replicación de S3 (S3 RTC)
- Métricas de solicitud de Amazon CloudWatch
- Configuración de métricas
- Transfer Acceleration
- Notificaciones de eventos de S3
- Los buckets de pago por solicitante
- S3 Select
- Eventos de AWS Lambda
- Server access logging (Registro de acceso del servidor)
- Solicitudes HTTP POST
- SOAP

- Acceso al sitio web

Requisitos de red de S3 en Outposts

- Para enrutar solicitudes a un punto de acceso de S3 en Outposts, debe crear y configurar un punto de conexión de S3 en Outposts. Los siguientes límites se aplican a los puntos de enlace de S3 en Outposts:
  - Cada nube virtual privada (VPC) en un Outpost puede tener un punto de conexión asociado y puede tener hasta 100 puntos de conexión por Outpost.
  - Se pueden asignar varios puntos de acceso al mismo punto de conexión.
  - Los puntos de conexión solo se pueden agregar a VPC con bloques de CIDR en los subespacios de los siguientes rangos de CIDR:
    - 10.0.0.0/8
    - 172.16.0.0/12
    - 192.168.0.0/16
- Los puntos de conexión de un Outpost solo se pueden crear a partir de las VPC que tengan bloques de CIDR no superpuestos.
- Solo se puede crear un punto de conexión desde su subred de Outposts.
- La subred utilizada para crear un punto de conexión debe contener cuatro direcciones IP para que S3 en Outposts pueda utilizarla.
- Si se especifica el grupo de direcciones IP propiedad del cliente (grupo de CoIP), este debe contener cuatro direcciones IP para que S3 en Outposts pueda utilizarlo.
- Solo puede crear un punto de conexión por Outpost por VPC.

Introducción a Amazon S3 en Outposts

Con Amazon S3 en Outposts, puede crear buckets de S3 en Outposts de AWS y almacenar y recuperar fácilmente objetos en las instalaciones para las aplicaciones que requieren acceso local a los datos, procesamiento local de los datos y residencia de los datos. S3 en Outposts proporciona una nueva clase de almacenamiento, S3 Outposts (`OUTPOSTS`), que utiliza las API de Amazon S3 y está diseñada para almacenar datos de manera duradera y redundante en múltiples dispositivos y servidores de AWS Outposts. Usted se comunica con su bucket de Outpost mediante un punto de acceso y una conexión de punto de conexión a través de una nube privada virtual (VPC). Puede usar las mismas API y características en los buckets de Outposts que en buckets de Amazon S3,

como políticas de acceso, cifrado y etiquetado. Puede utilizar S3 en Outposts a través de la AWS Management Console, AWS Command Line Interface (AWS CLI), AWS SDK o la API de REST.

Amazon S3 en Outposts le permite utilizar las API y las características de Amazon S3, como el almacenamiento de objetos, las políticas de acceso, el cifrado y el etiquetado, en AWS Outposts como lo hace en Amazon S3. Para obtener información sobre S3 en Outposts, consulte ¿Qué es Amazon S3 en Outposts?

Temas

- Configuración de IAM con S3 en Outposts
- Primeros pasos con AWS Management Console
- Introducción mediante AWS CLI y SDK para Java

Configuración de IAM con S3 en Outposts

AWS Identity and Access Management (IAM) es un Servicio de AWS que ayuda a los administradores a controlar de forma segura el acceso a los recursos de AWS. Los administradores de IAM controlan quién se puede autenticar (iniciar sesión) y autorizar (tener permisos) para utilizar los recursos de Amazon S3 en Outposts. IAM es un Servicio de AWS que se puede utilizar sin cargo adicional. De forma predeterminada, los usuarios no tienen permisos para los recursos y las operaciones de S3 en Outposts. Para conceder permisos de acceso para los recursos de S3 en Outposts y operaciones de API, puede usar IAM para crear usuarios, grupos o roles y adjuntar permisos.

Para dar acceso, agregue permisos a los usuarios, grupos o roles:

- Usuarios y grupos en AWS IAM Identity Center:

  Cree un conjunto de permisos. Siga las instrucciones de Creación de un conjunto de permisos en la Guía del usuario de AWS IAM Identity Center.
- Usuarios administrados en IAM a través de un proveedor de identidades:

  Cree un rol para la federación de identidades. Siga las instrucciones descritas en Creación de un rol para un proveedor de identidad de terceros (federación) en la Guía del usuario de IAM.
- Usuarios de IAM:
  - Cree un rol que el usuario pueda aceptar. Siga las instrucciones descritas en Creación de un rol para un usuario de IAM en la Guía del usuario de IAM.

- (No recomendado) Adjunte una política directamente a un usuario o añada un usuario a un grupo de usuarios. Siga las instrucciones descritas en Adición de permisos a un usuario (consola) de la Guía del usuario de IAM.

Además de las políticas de IAM basadas en identidad, S3 en Outposts admite políticas de punto de acceso y bucket. Las políticas de punto de acceso y bucket son políticas de basadas en recursos que están asociadas al recurso S3 en Outposts.

- Una política de bucket se asocia al bucket y permite o deniega solicitudes al bucket y a los objetos que hay en él en función de los elementos de la política.
- Por el contrario, se adjunta una política de punto de acceso al punto de acceso y permite o deniega solicitudes al punto de acceso.

La política de punto de acceso funciona con la política de bucket asociada al bucket S3 en Outposts subyacente. Para que una aplicación o un usuario pueda acceder a objetos en un bucket de S3 en Outposts a través de un punto de acceso de S3 en Outposts, tanto la política de punto de acceso como la política de bucket deben permitir la solicitud.

Las restricciones que se incluyen en una política de punto de acceso solo se aplican a las solicitudes realizadas a través de ese punto de acceso. Por ejemplo, si un punto de acceso está conectado a un bucket, no puede usar la política de punto de acceso para permitir o denegar las solicitudes que se realizan directamente en el bucket. Sin embargo, las restricciones que se aplican a una política de bucket pueden permitir o denegar solicitudes realizadas directamente al bucket o a través del punto de acceso.

En una política de IAM o una política basada en recursos, usted define qué acciones de S3 en Outposts se permiten o deniegan. Las acciones de S3 en Outposts corresponden a operaciones específicas de la API S3 en Outposts. Las acciones de S3 en Outposts utilizan el prefijo de espacio de nombres `s3-outposts:`. Las solicitudes realizadas a la API de control de S3 en Outposts en una Región de AWS y las solicitudes realizadas a los puntos de conexión de la API de objeto en el Outpost se autentican mediante IAM y se autorizan en el prefijo de espacio de nombres `s3-outposts:`. Para trabajar con S3 en Outposts, configure los usuarios de IAM y autorícelos en el espacio de nombres de IAM de `s3-outposts:`.

Para obtener información, consulte Acciones, recursos y claves de condición de Amazon S3 en Outposts en la Referencia de autorizaciones de servicio.

> **Note**
>
> - S3 en Outposts no admite las listas de control de acceso (ACL).
> - S3 en Outposts toma de forma predeterminada al propietario del bucket como propietario del objeto, para ayudar a garantizar que no se pueda impedir que el propietario de un bucket acceda a los objetos o los elimine.
> - S3 en Outposts siempre tiene Bloquear Acceso público en S3 habilitado para ayudar a garantizar que los objetos nunca puedan tener acceso público.

Para obtener más información acerca de la configuración de IAM para S3 en Outposts, consulte los siguientes temas.

Temas

Entidades principales para las políticas de S3 en Outposts

Cuando crea una política basada en recursos para conceder acceso a su bucket S3 en Outposts, debe utilizar el elemento `Principal` para especificar la persona o aplicación que puede realizar una solicitud para realizar una acción o una operación en ese recurso. Para las políticas S3 en Outposts, puede utilizar una de las siguientes entidades principales:

- Una Cuenta de AWS
- Un usuario de IAM
- Un rol de IAM
- Todas las entidades principales mediante la especificación de un carácter comodín (*) de una política que utiliza un elemento `Condition` para limitar el acceso a un rango de IP específicas

> **Important**
>
> No puede escribir una política para un bucket de S3 en Outposts que utilice un carácter comodín (`*`) en el elemento `Principal` a menos que la política también incluya una `Condition` que limite el acceso a un rango de direcciones IP específicas. Esta restricción contribuye a asegurar que no hay acceso público a su bucket de S3 en Outposts. Para ver un ejemplo, consulte Ejemplos de políticas para S3 en Outposts.

Para obtener más información acerca del elemento `Principal`, consulte Elementos de la política JSON de AWS: entidad principal en la Guía del usuario de IAM.

ARN de recursos para S3 en Outposts

Los nombres de recurso de Amazon (ARN) para S3 en Outposts contienen el ID de Outpost, además de la Región de AWS donde está destinado el Outpost, el ID de Cuenta de AWS y el nombre del recurso. Para acceder y realizar acciones en los buckets y objetos de Outposts, debe utilizar uno de los formatos de ARN que se muestran en la tabla siguiente.

El valor *`partition`* en el ARN hace referencia a un grupo de Regiones de AWS. Cada Cuenta de AWS está limitada a una partición. Las siguientes son las particiones admitidas:

- `aws` – Regiones de AWS
- `aws-us-gov`: regiones de AWS GovCloud (US)

Formatos de ARN de S3 en Outposts

ARN de Amazon S3 en Outposts	Formato de ARN	Ejemplo
ARN de bucket	`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/`*`amzn-s3-demo-bucket1`*

ARN de Amazon S3 en Outposts	Formato de ARN	Ejemplo
ARN del punto de acceso.	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 de objeto	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/*amzn-s3-demo-bucket1* /object/*myobject*
ARN de objeto de punto de acceso de S3 en Outposts (utilizado en políticas)	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 de S3 en Outposts	arn:*partition* :s3-outposts: *region*: *account_id* :outpost/ *outpost_id*	arn:*aws*:s3-outposts: *us-west-2* :*123456789012* :outpost/ *op-01ac5d28a6a232904*

Ejemplos de políticas para S3 en Outposts

Example : política de bucket de S3 en Outposts con una entidad principal de Cuenta de AWS

La siguiente política de bucket utiliza una entidad principal de Cuenta de AWS para conceder acceso a un bucket S3 en Outposts. Para utilizar esta política de bucket, reemplace *user input placeholders* por su propia información.

```
{
   "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 : política de bucket de S3 en Outposts con clave de entidad principal y condición (*) para limitar el acceso a un rango de direcciones IP específicas

La siguiente política de bucket utiliza una entidad principal comodín (*) con la condición `aws:SourceIp` para limitar el acceso a un rango de direcciones IP específicas. Para utilizar esta política de bucket, reemplace *user input placeholders* por su propia información.

```
{
    "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"
                }
            }
        }
    ]
}
```

Permisos para los puntos de conexión de S3 en Outposts

S3 en Outposts requiere sus propios permisos en IAM para administrar las acciones de puntos de conexión de S3 en Outposts.

> **Note**
>
> - Para los puntos de conexión que utilizan el tipo de acceso de grupo de direcciones IP propiedad del cliente (grupo de CoIP), también debe tener permisos para trabajar con direcciones IP desde el grupo de CoIP, como se describe en la siguiente tabla.
> - Para cuentas compartidas que acceden a S3 en Outposts mediante AWS Resource Access Manager, los usuarios en estas cuentas compartidas no pueden crear sus propios puntos de conexión en una subred compartida. Si el usuario de una cuenta compartida desea administrar sus propios puntos de conexión, la cuenta compartida debe crear su propia subred en Outpost. Para obtener más información, consulte the section called “Uso comaprtido de S3 en Outposts”.

Permisos de IAM relacionados con los puntos de conexión de S3 en Outposts

Acción	Permisos de IAM
`CreateEndpoint`	`s3-outposts:CreateEndpoint` `ec2:CreateNetworkInterface`

Acción	Permisos de IAM
	`ec2:DescribeNetworkInterfaces` `ec2:DescribeVpcs` `ec2:DescribeSecurityGroups` `ec2:DescribeSubnets` `ec2:CreateTags` `iam:CreateServiceLinkedRole` Para los puntos de enlace que utilizan el tipo de acceso de grupo de direcciones IP propiedad del cliente (grupo CoIP) en las instalaciones, se requieren los siguientes permisos adicionales: `s3-outposts:CreateEndpoint` `ec2:DescribeCoipPools` `ec2:GetCoipPoolUsage` `ec2:AllocateAddress` `ec2:AssociateAddress` `ec2:DescribeAddresses` `ec2:DescribeLocalGatewayRouteTableVpcAssociations`

Acción	Permisos de IAM
`DeleteEndpoint`	`s3-outposts:DeleteEndpoint` `ec2:DeleteNetworkInterface` `ec2:DescribeNetworkInterfaces` Para los puntos de enlace que utilizan el tipo de acceso de grupo de direcciones IP propiedad del cliente (grupo CoIP) en las instalaciones, se requieren los siguientes permisos adicionales: `s3-outposts:DeleteEndpoint` `ec2:DisassociateAddress` `ec2:DescribeAddresses` `ec2:ReleaseAddress`
`ListEndpoints`	`s3-outposts:ListEndpoints`

> **Note**
>
> Puede utilizar etiquetas de recursos en una política del IAM para administrar permisos.

Roles vinculados a servicios para S3 en Outposts

S3 en Outposts usa roles vinculados a servicios de IAM para crear algunos recursos de red en su nombre. Para obtener más información, consulte Uso de roles vinculados a servicios para Amazon S3 en Outposts.

Primeros pasos con AWS Management Console

Con Amazon S3 en Outposts, puede crear buckets de S3 en Outposts de AWS y almacenar y recuperar fácilmente objetos en las instalaciones para las aplicaciones que requieren acceso local

a los datos, procesamiento local de los datos y residencia de los datos. S3 en Outposts proporciona una nueva clase de almacenamiento, S3 Outposts (OUTPOSTS), que utiliza las API de Amazon S3 y está diseñada para almacenar datos de manera duradera y redundante en múltiples dispositivos y servidores de AWS Outposts. Usted se comunica con su bucket de Outpost mediante un punto de acceso y una conexión de punto de conexión a través de una nube privada virtual (VPC). Puede usar las mismas API y características en los buckets de Outposts que en buckets de Amazon S3, como políticas de acceso, cifrado y etiquetado. Puede utilizar S3 en Outposts a través de la AWS Management Console, AWS Command Line Interface (AWS CLI), AWS SDK o la API de REST. Para obtener más información, consulte ¿Qué es Amazon S3 en Outposts?

Para comenzar a utilizar S3 en Outposts mediante la consola, consulte los siguientes temas. Para comenzar a utilizar AWS CLI o AWS SDK for Java, consulte Introducción mediante AWS CLI y SDK para Java.

Temas

Cree un bucket, un punto de acceso y un punto de conexión

El siguiente procedimiento muestra cómo crear el primer bucket en S3 en Outposts. La primera vez que crea un bucket con la consola, también crea un punto de acceso y un punto de conexión asociados al bucket para que pueda comenzar a almacenar objetos inmediatamente en el bucket.

1. Inicie sesión en la AWS Management Console y abra la consola de Amazon S3 en https://console.aws.amazon.com/s3/.
2. En el panel de navegación de la izquierda, elija Outposts buckets (Buckets de Outposts).
3. Seleccione Crear bucket de Outposts.
4. En Bucket name (Nombre del bucket), escriba un nombre compatible con sistema de nombres de dominio (DNS) para el bucket.

   El nombre del bucket debe:

   - Ser único dentro de la Cuenta de AWS, el Outpost y la Región de AWS al que está destinado el Outpost.
   - Tener de 3 a 63 caracteres.
   - No contiene caracteres en mayúsculas.

- Comenzar por una letra minúscula o un número.

  Una vez que haya creado el bucket, no podrá modificar su nombre. Para obtener información acerca de la nomenclatura de los buckets, consulte Reglas de nomenclatura de buckets.

  > **Important**
  >
  > Evite incluir información confidencial, como números de cuenta, en el nombre del bucket. El nombre del bucket será visible en las URL que señalan a los objetos almacenados en él.

5. En Outposts, elija el Outpost donde desea que resida el bucket.
6. En Bucket Versioning (Control de versiones de bucket), establezca el estado de control de versiones de S3 para su bucket de S3 on Outposts en una de las siguientes opciones:

   - Disable (Deshabilitar) (predeterminado): el bucket permanece sin versiones.
   - Enable (Habilitar): habilita el control de versiones de S3 para los objetos del bucket. Todos los objetos añadidos al bucket reciben un ID de versión único.

   Para obtener más información sobre el control de versiones de S3, consulte Administración de control de versiones de S3 para su bucket de S3 en Outposts.
7. (Opcional) Agregue las etiquetas opcionales que desee asociar con el bucket de Outposts. Puede usar etiquetas para realizar un seguimiento de los criterios para proyectos individuales o grupos de proyectos o para etiquetar los buckets con etiquetas de asignación de costos.

   De manera predeterminada, todos los objetos almacenados en el bucket de Outposts se almacenan mediante cifrado del lado del servidor con claves de cifrado administradas por Amazon S3 (SSE-S3). También puede elegir almacenar objetos mediante cifrado del lado del servidor con claves de cifrado proporcionadas por el cliente (SSE-C). Para cambiar el tipo de cifrado, debe utilizar la API de REST, AWS Command Line Interface (AWS CLI) o SDK de AWS.
8. En la sección Configuración del punto de acceso de Outposts, introduzca el nombre del punto de acceso.

   Los puntos de acceso de S3 en Outposts simplifican la administración del acceso a los datos a escala para los conjuntos de datos compartidos en S3 en Outposts. Los puntos de acceso son puntos de enlace de red con nombre que están asociados a los buckets Outposts que se

pueden utilizar para realizar operaciones con objetos de S3. Para obtener más información, consulte Puntos de acceso.

Los nombres de los puntos de acceso deben ser únicos dentro de la cuenta para esta región y Outposts, y cumplir con Restricciones y limitaciones de los puntos de acceso.

9. Elija la VPC para este punto de acceso de Amazon S3 en Outposts.

   Si no tiene una VPC, elija Create VPC (Crear VPC). Para obtener más información, consulte Crear puntos de acceso restringidos a una nube privada virtual.

   Una nube virtual privada (VPC) le permite lanzar recursos de AWS en una red virtual que defina. Dicha red virtual es prácticamente idéntica a las redes tradicionales que se utilizarían en sus propios centros de datos, con los beneficios que supone utilizar la infraestructura escalable de AWS.

10. (Opcional para una VPC existente) Elija una Subred de punto de conexión para el punto de conexión.

    Una subred es un rango de direcciones IP en su VPC. Si no tiene la subred que desea, elija Create subnet (Crear subred). Para obtener más información, consulte Redes para S3 en Outposts.

11. (Opcional para una VPC existente) Elija un Grupo de seguridad de puntos de conexión para el punto de conexión.

    Un grupo de seguridad funciona como un firewall virtual para controlar el tráfico entrante y saliente.

12. (Opcional para una VPC existente) Elija el Endpoint access type (Tipo de acceso al punto de conexión):

    - Privado: para utilizarse con la VPC.
    - IP de propiedad del cliente: se utiliza con un grupo de direcciones IP (grupo CoIP) desde la red de las instalaciones.

13. (Opcional) Especifique la Outpost access point policy (Política de punto de acceso de Outpost). La consola muestra automáticamente el nombre de recurso de Amazon (ARN) para el punto de acceso, que puede utilizar en la política.

14. Seleccione Crear bucket de Outposts.

> **Note**
>
> Puede tardar hasta 5 minutos para que se cree el punto de conexión de Outpost y se pueda usar el bucket. Para configurar opciones adicionales de bucket, elija View details (Ver detalles).

Pasos siguientes

Con Amazon S3 en Outposts, los datos de objeto siempre se almacenan en el Outpost. Cuando AWS instala un bastidor de Outpost, sus datos permanecen de manera local en su Outpost para cumplir los requisitos de residencia de datos. Sus objetos nunca salen de su Outpost y no están en una Región de AWS. Ya que la AWS Management Console está alojada dentro de la región, no puede usar la consola para cargar o administrar objetos en su Outpost. Sin embargo, puede utilizar la API de REST, AWS Command Line Interface (AWS CLI) y los SDK de AWS para cargar y administrar los objetos a través de los puntos de acceso.

Después de crear un bucket de S3 en Outposts, un punto de acceso y un punto de conexión, puede utilizar la AWS CLI o el SDK para Java para cargar un objeto en el bucket. Para obtener más información, consulte Carga de un objeto en un bucket de S3 en Outpost.

Introducción mediante AWS CLI y SDK para Java

Con Amazon S3 en Outposts, puede crear buckets de S3 en Outposts de AWS y almacenar y recuperar fácilmente objetos en las instalaciones para las aplicaciones que requieren acceso local a los datos, procesamiento local de los datos y residencia de los datos. S3 en Outposts proporciona una nueva clase de almacenamiento, S3 Outposts (OUTPOSTS), que utiliza las API de Amazon S3 y está diseñada para almacenar datos de manera duradera y redundante en múltiples dispositivos y servidores de AWS Outposts. Usted se comunica con su bucket de Outpost mediante un punto de acceso y una conexión de punto de conexión a través de una nube privada virtual (VPC). Puede usar las mismas API y características en los buckets de Outposts que en buckets de Amazon S3, como políticas de acceso, cifrado y etiquetado. Puede utilizar S3 en Outposts a través de la AWS Management Console, AWS Command Line Interface (AWS CLI), AWS SDK o la API de REST. Para obtener más información, consulte ¿Qué es Amazon S3 en Outposts?.

Para empezar a utilizar S3 en Outposts, debe crear un bucket, un punto de acceso y un punto de conexión. Luego, puede cargar objetos en el bucket. Los siguientes ejemplos muestran cómo puede

utilizar S3 en Outposts mediante AWS CLI y el SDK para Java. Para comenzar mediante la consola, consulte Primeros pasos con AWS Management Console.

Temas

Paso 1: Crear un bucket

Los siguientes ejemplos de AWS CLI y SDK para Java muestran cómo puede crear un bucket de S3 en Outposts.

AWS CLI

Example

En el siguiente ejemplo, se crea un bucket de S3 en Outposts (`s3-outposts:CreateBucket`) con la AWS CLI. Para ejecutar este comando, sustituya los *`user input placeholders`* con su propia información.

```
aws s3control create-bucket --bucket example-outposts-bucket --outpost-
id op-01ac5d28a6a232904
```

SDK for Java

Example

En el siguiente ejemplo, se crea un bucket de S3 en Outposts (`s3-outposts:CreateBucket`) con el SDK para 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();

}
```

Paso 2: Crear un punto de acceso

Para acceder a su bucket de Amazon S3 en Outposts, debe crear y configurar un punto de acceso. En estos ejemplos, se explica cómo crear un punto de acceso mediante AWS CLI y el SDK para Java.

Los puntos de acceso simplifican la administración del acceso a los datos a escala para los conjuntos de datos compartidos en Amazon S3. Los puntos de acceso son puntos de enlace de red con nombre y asociados a los buckets que se pueden utilizar para realizar operaciones con objetos de Amazon S3, como `GetObject` y `PutObject`. Con S3 en Outposts, debe utilizar puntos de acceso para acceder a cualquier objeto de un bucket de Outposts. Los puntos de acceso solo admiten el direccionamiento de tipo de host virtual.

AWS CLI

Example

En el siguiente ejemplo de la AWS CLI, se crea un punto de acceso para un bucket de Outposts. Para ejecutar este comando, sustituya los *`user input placeholders`* con su propia información.

```
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

En el siguiente ejemplo del SDK para Java, se crea un punto de acceso para un bucket de Outposts. Para utilizar este ejemplo, reemplace los *user input placeholders* con su propia información.

```
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();

}
```

Paso 3: Crear un punto de conexión

Para dirigir solicitudes a un punto de acceso de Amazon S3 en Outposts, debe crear y configurar un punto de conexión de S3 en Outposts. Para crear un punto de conexión, necesita una conexión activa con el enlace de servicio a la región de origen de Outposts. Cada nube virtual privada (VPC) de su Outpost puede tener un punto de conexión asociado. Para obtener más información acerca de las cuotas de los puntos de conexión, consulte Requisitos de red de S3 en Outposts. Debe crear un punto de conexión para poder acceder a los buckets de Outposts y realizar operaciones de objetos. Para obtener más información, consulte Puntos de conexión.

En estos ejemplos, se muestra cómo crear un punto de conexión mediante AWS CLI y el SDK para Java. Para obtener más información acerca de los permisos necesarios para crear y administrar puntos de conexión, consulte Permisos para los puntos de conexión de S3 en Outposts.

AWS CLI

Example

En el siguiente ejemplo de la AWS CLI, se crea un punto de conexión para un Outpost con el tipo de acceso a recursos de la VPC. La VPC se obtiene de la subred. Para ejecutar este comando, sustituya los *user input placeholders* con su propia información.

```
aws s3outposts create-endpoint --outpost-id op-01ac5d28a6a232904 --subnet-id
 subnet-8c7a57c5 --security-group-id sg-ab19e0d1
```

En el siguiente ejemplo de la AWS CLI, se crea un punto de conexión para un Outpost con el tipo de acceso de grupo de direcciones IP propiedad del cliente (grupo de CoIP). Para ejecutar este comando, sustituya los *user input placeholders* con su propia información.

```
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

En el siguiente ejemplo del SDK para Java, se crea un punto de conexión para un Outpost. Para utilizar este ejemplo, reemplace los *user input placeholders* con su propia información.

```
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());
}
```

Paso 4: Cargar un objeto en un bucket de S3 en Outposts

Para cargar un objeto, consulte Carga de un objeto en un bucket de S3 en Outpost.

Redes para S3 en Outposts

Puede utilizar Amazon S3 en Outposts para almacenar y recuperar objetos locales para aplicaciones que requieren acceso a datos locales, procesamiento de datos y residencia de datos. En esta sección se describen los requisitos de red para acceder a S3 en Outposts.

Temas

- Elección del tipo de acceso de red
- Acceso a los buckets y objetos de S3 en Outposts
- Interfaces de red elástica entre cuentas

Elección del tipo de acceso de red

Puede acceder a S3 en Outposts desde una VPC o desde su red en las instalaciones. Usted se comunica con su bucket de Outpost mediante un punto de acceso y una conexión de punto de conexión. Esta conexión mantiene el tráfico entre su VPC y sus buckets S3 en Outposts dentro de la red de AWS. Cuando se crea un punto de conexión, se debe especificar el tipo de acceso de punto de conexión entre `Private` (para enrutamiento VPC) o `CustomerOwnedIp` (para grupo de dirección IP propiedad del cliente [grupo CoIP]).

- `Private` (para el enrutamiento de VPC): si no se especifica el tipo de acceso, S3 en Outposts se utiliza `Private` de forma predeterminada. Con el tipo de acceso `Private`, las instancias de su VPC no necesitan direcciones IP públicas para comunicarse con los recursos de su Outposts. Puede trabajar con S3 en Outposts desde una VPC. Este tipo de punto de conexión es accesible

desde la red en las instalaciones a través del enrutamiento directo de VPC. Para obtener más información, consulte Tablas de enrutamiento de puerta de enlace locales en la Guía del usuario de AWS Outposts.

- `CustomerOwnedIp` (para el grupo de CoIP): si no se utiliza de forma predeterminada el tipo de acceso `Private` y elige `CustomerOwnedIp`, debe especificar un rango de direcciones IP. Puede usar este tipo de acceso para trabajar con S3 en Outposts desde la red de las instalaciones y en VPC. Al acceder a S3 en Outposts dentro de una VPC, su tráfico está limitado a la banda ancha de la gateway local.

Acceso a los buckets y objetos de S3 en Outposts

Para acceder a sus cubos y objetos de S3 en Outposts, debe tener lo siguiente:

- Un punto de acceso para VPC
- Un punto de enlace para la misma VPC
- Una conexión activa entre Outpost y su Región de AWS. Para obtener más información sobre cómo conectar su Outpost con una región, consulte Conectividad de Outpost con regiones de AWS en la AWSGuía de usuario de Outposts.

Para obtener más información acerca de cómo acceder a buckets y objetos en S3 en Outposts, consulte Trabajo con buckets de S3 en Outposts y Trabajo con objetos de S3 en Outposts.

Interfaces de red elástica entre cuentas

Los puntos de conexión de S3 en Outposts son recursos designados con nombres de recurso de Amazon (ARN). Cuando se crean estos puntos de enlace, AWS Outposts configura cuatro interfaces de red elástica entre cuentas. Las interfaces de red elástica entre cuentas de S3 en Outposts son como otras interfaces de red con una excepción: S3 en Outposts asocia las interfaces de red elásticas entre cuentas con instancias de Amazon EC2.

La carga del sistema de nombres de dominio (DNS) de S3 en Outposts equilibra las solicitudes sobre la interfaz de red elástica entre cuentas. S3 en Outposts crea la interfaz de red elástica entre cuentas en su cuenta de AWS que es visible desde el panel Network interfaces (Interfaces de red) de la consola de Amazon EC2.

Para los puntos de enlace que utilizan el tipo de acceso de grupo de CoIP, S3 en Outposts asigna y asocia direcciones IP a la interfaz de red elástica entre cuentas desde el grupo de CoIP configurado.

Trabajo con buckets de S3 en Outposts

Con Amazon S3 en Outposts, puede crear buckets de S3 en AWS Outposts y almacenar y recuperar fácilmente objetos en las instalaciones para las aplicaciones que requieren acceso local a los datos, procesamiento local de los datos y residencia de los datos. S3 en Outposts proporciona una nueva clase de almacenamiento, S3 Outposts (OUTPOSTS), que utiliza las API de Amazon S3 y está diseñada para almacenar datos de manera duradera y redundante en múltiples dispositivos y servidores de AWS Outposts. Puede usar las mismas API y características en los buckets de Outpost que en Amazon S3, como políticas de acceso, cifrado y etiquetado. Para obtener más información, consulte ¿Qué es Amazon S3 en Outposts?

Usted se comunica con sus buckets de Outpost mediante un punto de acceso y una conexión de punto de conexión a través de una nube privada virtual (VPC). Para acceder a sus buckets y objetos de S3 en Outposts, debe tener un punto de acceso para la VPC y un punto de conexión para la misma VPC. Para obtener más información, consulte Redes para S3 en Outposts.

Buckets

En S3 en Outposts, los nombres de bucket son únicos de un Outpost y requieren el código de Región de AWS para la región a la que está destinado el Outpost, el ID de Cuenta de AWS, el ID de Outpost y el nombre del bucket para identificarlos.

```
arn:aws:s3-outposts:region:account-id:outpost/outpost-id/bucket/bucket-name
```

Para obtener más información, consulte ARN de recursos para S3 en Outposts.

Puntos de acceso

Amazon S3 en Outposts admite puntos de acceso únicamente de la virtual private cloud (VPC) como el único medio para acceder a los buckets de Outposts.

Los puntos de acceso simplifican la administración del acceso a los datos a escala para los conjuntos de datos compartidos en Amazon S3. Los puntos de acceso son puntos de enlace de red con nombre y asociados a los buckets que se pueden utilizar para realizar operaciones con objetos de Amazon S3, como `GetObject` y `PutObject`. Con S3 en Outposts, debe utilizar puntos de acceso para acceder a cualquier objeto de un bucket de Outposts. Los puntos de acceso solo admiten el direccionamiento de tipo de host virtual.

En el siguiente ejemplo, se muestra el formato ARN que se utiliza para los puntos de acceso de S3 en Outposts. El ARN de punto de acceso incluye el código de Región de AWS para la región a la que está destinado el Outpost, el ID de Cuenta de AWS, el ID de Outpost y el nombre de punto de acceso.

```
arn:aws:s3-outposts:region:account-id:outpost/outpost-id/accesspoint/accesspoint-name
```

Puntos de conexión

Para enrutar solicitudes a un punto de acceso de S3 en Outposts, debe crear y configurar un punto de conexión de S3 en Outposts. Con los puntos de conexión de S3 en Outposts, puede conectar de forma privada su VPC al bucket de Outpost. Los puntos de enlace de S3 en Outposts son identificadores de recursos uniformes (URI) virtuales del punto de entrada al bucket de S3 en Outposts. Son componentes de VPC escalados horizontalmente, redundantes y de alta disponibilidad.

Cada nube virtual privada (VPC) en su Outpost puede tener un punto de conexión asociado y puede tener hasta 100 puntos de conexión por Outpost. Debe crear estos puntos de conexión para poder acceder a los buckets de Outpost y realizar operaciones de objetos. De esta forma, también hace que el modelo y los comportamientos de la API sean los mismos al permitir que las mismas operaciones funcionen en S3 y S3 en Outposts.

Operaciones de API en S3 en Outposts

S3 en Outposts aloja un punto de conexión separado para administrar las operaciones de API de bucket de Outposts distintas de los puntos de conexión de Amazon S3. Este punto de enlace es `s3-outposts.region.amazonaws.com`.

Para utilizar las operaciones de la API de Amazon S3, debe firmar el bucket y los objetos con el formato ARN correcto. Debe pasar ARN para las operaciones de API a fin de que Amazon S3 pueda determinar si la solicitud es para Amazon S3 (`s3-control.region.amazonaws.com`) o S3 en Outposts (`s3-outposts.region.amazonaws.com`). Según el formato ARN, luego S3 puede firmar y dirigir la solicitud de forma adecuada.

Siempre que la solicitud se envía al plano de control de Amazon S3, el SDK extrae los componentes del ARN e incluye un encabezado adicional `x-amz-outpost-id` con el valor de *outpost-id* que se extrajo del ARN. El nombre del servicio del ARN se utiliza para firmar la solicitud antes de que se dirija al punto de enlace de S3 en Outposts. Este comportamiento se aplica a todas las operaciones de API manejadas por el cliente `s3control`.

En la siguiente tabla, se enumeran las operaciones API ampliadas para Amazon S3 en Outposts y sus cambios en relación con Amazon S3.

API	Valor del parámetro de S3 en Outposts	
`CreateBucket`	Nombre del bucket como ARN, ID de Outpost	
`ListRegionalBuckets`	ID de Outpost	
`DeleteBucket`	Nombre del bucket como ARN	
`DeleteBucketLifecycleConfiguration`	Nombre del bucket como ARN	
`GetBucketLifecycleConfiguration`	Nombre del bucket como ARN	
`PutBucketLifecycleConfiguration`	Nombre del bucket como ARN	
`GetBucketPolicy`	Nombre del bucket como ARN	
`PutBucketPolicy`	Nombre del bucket como ARN	
`DeleteBucketPolicy`	Nombre del bucket como ARN	
`GetBucketTagging`	Nombre del bucket como ARN	
`PutBucketTagging`	Nombre del bucket como ARN	
`DeleteBucketTagging`	Nombre del bucket como ARN	
`CreateAccessPoint`	Nombre del punto de acceso como ARN	
`DeleteAccessPoint`	Nombre del punto de acceso como ARN	

API	Valor del parámetro de S3 en Outposts	
`GetAccessPoint`	Nombre del punto de acceso como ARN	
`GetAccessPoint`	Nombre del punto de acceso como ARN	
`ListAccessPoints`	Nombre del punto de acceso como ARN	
`PutAccessPointPolicy`	Nombre del punto de acceso como ARN	
`GetAccessPointPolicy`	Nombre del punto de acceso como ARN	
`DeleteAccessPointPolicy`	Nombre del punto de acceso como ARN	

Creación y administración de buckets de S3 en Outposts

Para obtener más información acerca de cómo crear y administrar los buckets de S3 en Outposts, consulte los siguientes temas.

Temas

- Creación de un bucket de S3 en Outposts
- Agregar etiquetas para los buckets de S3 en Outposts
- Administración del acceso a su bucket de Amazon S3 en Outposts mediante una política de bucket
- Obtención de una lista de buckets de Amazon S3 en Outposts
- Obtención de un bucket de S3 en Outposts mediante la AWS CLI y el SDK para Java
- Eliminación del bucket de Amazon S3 en Outposts
- Trabajo con puntos de acceso de Amazon S3 en Outposts
- Trabajo con puntos de conexión de Amazon S3 en Outposts

Creación de un bucket de S3 en Outposts

Con Amazon S3 en Outposts, puede crear buckets de S3 en AWS Outposts y almacenar y recuperar fácilmente objetos en las instalaciones para las aplicaciones que requieren acceso local a los datos, procesamiento local de los datos y residencia de los datos. S3 en Outposts proporciona una nueva clase de almacenamiento, S3 Outposts (OUTPOSTS), que utiliza las API de Amazon S3 y está diseñada para almacenar datos de manera duradera y redundante en múltiples dispositivos y servidores de AWS Outposts. Usted se comunica con su bucket de Outpost mediante un punto de acceso y una conexión de punto de conexión a través de una nube privada virtual (VPC). Puede usar las mismas API y características en los buckets de Outposts que en buckets de Amazon S3, como políticas de acceso, cifrado y etiquetado. Puede utilizar S3 en Outposts a través de la AWS Management Console, AWS Command Line Interface (AWS CLI), AWS SDK o la API de REST. Para obtener más información, consulte ¿Qué es Amazon S3 en Outposts?.

> **Note**
>
> La Cuenta de AWS que crea el bucket es su propietaria y la única que puede confirmarle acciones. Los buckets tienen propiedades de configuración como Outpost, etiquetas, cifrado predeterminado y valores de puntos de acceso. La configuración de punto de acceso incluye la VPC (nube virtual privada) y la política de punto de acceso para acceder a los objetos del bucket y otros metadatos. Para obtener más información, consulte Especificaciones de S3 en Outposts.
>
> Si desea crear un bucket que utilice AWS PrivateLink para proporcionar acceso a la administración de buckets y puntos de conexión a través de puntos de conexión de VPC de la interfaz en su nube privada virtual (VPC), consulte AWS PrivateLink para S3 en Outposts.

En los siguientes ejemplos se muestra cómo crear un bucket de S3 en Outposts con la AWS Management Console, AWS Command Line Interface (AWS CLI) y AWS SDK for Java.

Uso de la consola de S3

1. Inicie sesión en la AWS Management Console y abra la consola de Amazon S3 en https://console.aws.amazon.com/s3/.
2. En el panel de navegación de la izquierda, elija Outposts buckets (Buckets de Outposts).
3. Seleccione Crear bucket de Outposts.

4. En Bucket name (Nombre del bucket), escriba un nombre compatible con sistema de nombres de dominio (DNS) para el bucket.

   El nombre del bucket debe:

   - Ser único dentro de la Cuenta de AWS, el Outpost y la Región de AWS al que está destinado el Outpost.
   - Tener de 3 a 63 caracteres.
   - No contiene caracteres en mayúsculas.
   - Comenzar por una letra minúscula o un número.

   Una vez que haya creado el bucket, no podrá modificar su nombre. Para obtener información acerca de la nomenclatura de los buckets, consulte Reglas de nomenclatura de buckets.

   > **Important**
   >
   > Evite incluir información confidencial, como números de cuenta, en el nombre del bucket. El nombre del bucket será visible en las URL que señalan a los objetos almacenados en él.

5. En Outposts, elija el Outpost donde desea que resida el bucket.
6. En Bucket Versioning (Control de versiones de bucket), establezca el estado de control de versiones de S3 para su bucket de S3 on Outposts en una de las siguientes opciones:

   - Disable (Deshabilitar) (predeterminado): el bucket permanece sin versiones.
   - Enable (Habilitar): habilita el control de versiones de S3 para los objetos del bucket. Todos los objetos añadidos al bucket reciben un ID de versión único.

   Para obtener más información sobre el control de versiones de S3, consulte Administración de control de versiones de S3 para su bucket de S3 en Outposts.
7. (Opcional) Agregue las etiquetas opcionales que desee asociar con el bucket de Outposts. Puede usar etiquetas para realizar un seguimiento de los criterios para proyectos individuales o grupos de proyectos o para etiquetar los buckets con etiquetas de asignación de costos.

   De manera predeterminada, todos los objetos almacenados en el bucket de Outposts se almacenan mediante cifrado del lado del servidor con claves de cifrado administradas por Amazon S3 (SSE-S3). También puede elegir almacenar objetos mediante cifrado del lado del

servidor con claves de cifrado proporcionadas por el cliente (SSE-C). Para cambiar el tipo de cifrado, debe utilizar la API de REST, AWS Command Line Interface (AWS CLI) o SDK de AWS.

8. En la sección Configuración del punto de acceso de Outposts, introduzca el nombre del punto de acceso.

   Los puntos de acceso de S3 en Outposts simplifican la administración del acceso a los datos a escala para los conjuntos de datos compartidos en S3 en Outposts. Los puntos de acceso son puntos de enlace de red con nombre que están asociados a los buckets Outposts que se pueden utilizar para realizar operaciones con objetos de S3. Para obtener más información, consulte Puntos de acceso.

   Los nombres de los puntos de acceso deben ser únicos dentro de la cuenta para esta región y Outposts, y cumplir con Restricciones y limitaciones de los puntos de acceso.

9. Elija la VPC para este punto de acceso de Amazon S3 en Outposts.

   Si no tiene una VPC, elija Create VPC (Crear VPC). Para obtener más información, consulte Crear puntos de acceso restringidos a una nube privada virtual.

   Una nube virtual privada (VPC) le permite lanzar recursos de AWS en una red virtual que defina. Dicha red virtual es prácticamente idéntica a las redes tradicionales que se utilizarían en sus propios centros de datos, con los beneficios que supone utilizar la infraestructura escalable de AWS.

10. (Opcional para una VPC existente) Elija una Subred de punto de conexión para el punto de conexión.

    Una subred es un rango de direcciones IP en su VPC. Si no tiene la subred que desea, elija Create subnet (Crear subred). Para obtener más información, consulte Redes para S3 en Outposts.

11. (Opcional para una VPC existente) Elija un Grupo de seguridad de puntos de conexión para el punto de conexión.

    Un grupo de seguridad funciona como un firewall virtual para controlar el tráfico entrante y saliente.

12. (Opcional para una VPC existente) Elija el Endpoint access type (Tipo de acceso al punto de conexión):

    - Privado: para utilizarse con la VPC.

- IP de propiedad del cliente: se utiliza con un grupo de direcciones IP (grupo CoIP) desde la red de las instalaciones.

13. (Opcional) Especifique la Outpost access point policy (Política de punto de acceso de Outpost). La consola muestra automáticamente el nombre de recurso de Amazon (ARN) para el punto de acceso, que puede utilizar en la política.
14. Seleccione Crear bucket de Outposts.

> **Note**
>
> Puede tardar hasta 5 minutos para que se cree el punto de conexión de Outpost y se pueda usar el bucket. Para configurar opciones adicionales de bucket, elija View details (Ver detalles).

Utilización de la AWS CLI

Example

En el siguiente ejemplo, se crea un bucket de S3 en Outposts (`s3-outposts:CreateBucket`) con la AWS CLI. Para ejecutar este comando, sustituya los *`user input placeholders`* con su propia información.

```
aws s3control create-bucket --bucket example-outposts-bucket --outpost-
id op-01ac5d28a6a232904
```

Uso de AWS SDK para Java

Example

En el siguiente ejemplo, se crea un bucket de S3 en Outposts (`s3-outposts:CreateBucket`) con el SDK para 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();

}
```

Agregar etiquetas para los buckets de S3 en Outposts

Puede agregar etiquetas para los buckets de Amazon S3 en Outposts para realizar un seguimiento de los costos de almacenamiento y otros criterios para proyectos individuales o grupos de proyectos.

> **Note**
>
> La Cuenta de AWS que crea el bucket es su propietaria y la única que puede cambiar sus etiquetas.

Uso de la consola de S3

1. Inicie sesión en la AWS Management Console y abra la consola de Amazon S3 en https://console.aws.amazon.com/s3/.
2. En el panel de navegación izquierdo, elija Outposts buckets (buckets de Outposts).
3. Elija el bucket de Outposts cuyas etiquetas desea editar.
4. Elija la pestaña Properties (Propiedades).
5. En Tags (Etiquetas), elija Edit (Editar).
6. Elija Add new tag (Agregar nueva etiqueta) e introduzca la clave y el valor opcional.

   Agregue las etiquetas que desee asociar con un bucket de Outposts para realizar un seguimiento de otros criterios para proyectos individuales o grupos de proyectos.
7. Elija Save changes.

Mediante AWS CLI

El siguiente ejemplo de AWS CLI aplica una configuración de etiquetado a un bucket de S3 en Outposts mediante un documento JSON de la carpeta actual que especifica etiquetas

(*tagging.json*). Para utilizar este ejemplo, sustituya *user input placeholder* por su propia información.

```
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"
     }
   ]
}
```

El siguiente ejemplo de AWS CLI aplica una configuración de etiquetado a un bucket de S3 en Outposts directamente desde la línea de comandos.

```
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}]'
```

Para obtener más información acerca de este comando, consulte put-bucket-tagging en la Referencia de AWS CLI.

Administración del acceso a su bucket de Amazon S3 en Outposts mediante una política de bucket

Una política de bucket es una política AWS Identity and Access Management basada en recursos (IAM) que puede utilizar para conceder permisos de acceso al bucket y a los objetos que contiene. Solo el propietario del bucket puede asociar una política a un bucket. Los permisos asociados a un bucket se aplican a todos los objetos del bucket que son propiedad de la cuenta de propietario del bucket. Las políticas de bucket tienen un límite de tamaño de 20 KB. Para obtener más información, consulte Política de bucket.

Puede actualizar la política de bucket para administrar el acceso a su bucket de Amazon S3 en Outposts. Para obtener más información, consulte los siguientes temas.

Temas

Adición o edición de una política de bucket para un bucket de Amazon S3 en Outposts

Una política de bucket es una política AWS Identity and Access Management basada en recursos (IAM) que puede utilizar para conceder permisos de acceso al bucket y a los objetos que contiene. Solo el propietario del bucket puede asociar una política a un bucket. Los permisos asociados a un bucket se aplican a todos los objetos del bucket que son propiedad de la cuenta de propietario del bucket. Las políticas de bucket tienen un límite de tamaño de 20 KB. Para obtener más información, consulte Política de bucket .

En los siguientes temas, se le mostrará cómo actualizar su política de bucket de Amazon S3 en Outposts mediante la AWS Management Console, AWS Command Line Interface (AWS CLI) o AWS SDK for Java.

Uso de la consola de S3

Para crear o editar una política de bucket

1. Inicie sesión en la AWS Management Console y abra la consola de Amazon S3 en https://console.aws.amazon.com/s3/.
2. En el panel de navegación de la izquierda, elija Outposts buckets (Buckets de Outposts).
3. Elija el bucket de Outposts cuya política de bucket desea editar.
4. Elija la pestaña Permissions (Permisos).
5. En la sección Outposts bucket policy (Política del bucket de Outposts), para crear o editar una nueva política, elija Edit (Editar).

   Ahora puede agregar o editar la política de bucket S3 en Outposts. Para obtener más información, consulte Configuración de IAM con S3 en Outposts .

Mediante AWS CLI

En el siguiente ejemplo de la AWS CLI, se aplica una política en un bucket de Outposts.

1. Guarde la política de bucket siguiente en un archivo JSON. En este ejemplo, el archivo se denomina `policy1.json`. Reemplace los *`user input placeholders`* con su propia información.

```
{
   "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. Envíe el archivo JSON como parte del comando de la CLI `put-bucket-policy`. Para ejecutar este comando, sustituya los *`user input placeholders`* con su propia información.

```
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
```

Uso de AWS SDK para Java

En el siguiente ejemplo del SDK para Java, se aplica una política en un bucket de 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());

}
```

Visualización de la política de bucket para el bucket de Amazon S3 en Outposts

Una política de bucket es una política AWS Identity and Access Management basada en recursos (IAM) que puede utilizar para conceder permisos de acceso al bucket y a los objetos que contiene. Solo el propietario del bucket puede asociar una política a un bucket. Los permisos asociados a un bucket se aplican a todos los objetos del bucket que son propiedad de la cuenta de propietario del bucket. Las políticas de bucket tienen un límite de tamaño de 20 KB. Para obtener más información, consulte Política de bucket .

En los siguientes temas, se muestra cómo ver la política de bucket de Amazon S3 en Outposts mediante la AWS Management Console, AWS Command Line Interface (AWS CLI) o AWS SDK for Java.

Uso de la consola de S3

Para crear o editar una política de bucket

1. Inicie sesión en la AWS Management Console y abra la consola de Amazon S3 en https://console.aws.amazon.com/s3/.
2. En el panel de navegación de la izquierda, elija Outposts buckets (Buckets de Outposts).
3. Elija el bucket de Outposts cuyo permiso desea editar.
4. Elija la pestaña Permissions.
5. En la sección Outposts bucket policy (Política de bucket de Outposts), puede revisar su política de bucket existente. Para obtener más información, consulte Configuración de IAM con S3 en Outposts .

Mediante AWS CLI

En el siguiente ejemplo de la AWS CLI, se obtiene una política para un bucket de Outposts. Para ejecutar este comando, sustituya los *user input placeholders* con su propia información.

```
aws s3control get-bucket-policy --account-id 123456789012 --bucket arn:aws:s3-
outposts:region:123456789012:outpost/op-01ac5d28a6a232904/bucket/example-outposts-
bucket
```

Uso de AWS SDK para Java

En el siguiente ejemplo del SDK para Java, se obtiene una política para un bucket de 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());

}
```

Eliminación de la política de bucket para su bucket de Amazon S3 en Outposts

Una política de bucket es una política AWS Identity and Access Management basada en recursos (IAM) que puede utilizar para conceder permisos de acceso al bucket y a los objetos que contiene. Solo el propietario del bucket puede asociar una política a un bucket. Los permisos asociados a un bucket se aplican a todos los objetos del bucket que son propiedad de la cuenta de propietario del bucket. Las políticas de bucket tienen un límite de tamaño de 20 KB. Para obtener más información, consulte Política de bucket .

En los siguientes temas, se muestra cómo ver la política de bucket de Amazon S3 en Outposts mediante la AWS Management Console o AWS Command Line Interface (AWS CLI).

Uso de la consola de S3

Para eliminar una política de bucket

1. Abra la consola de Amazon S3 en https://console.aws.amazon.com/s3.
2. En el panel de navegación de la izquierda, elija Outposts buckets (Buckets de Outposts).
3. Elija el bucket de Outposts cuyo permiso desea editar.
4. Elija la pestaña Permissions.
5. En la sección Outposts bucket policy (Política de bucket de Outposts), seleccione Delete (Eliminar).
6. Confirme la eliminación.

Mediante AWS CLI

En el siguiente ejemplo, se elimina la política de bucket para un bucket de S3 en Outposts (`s3-outposts:DeleteBucket`) utilizando la AWS CLI. Para ejecutar este comando, sustituya los *`user input placeholders`* con su propia información.

```
aws s3control delete-bucket-policy --account-id 123456789012 --bucket arn:aws:s3-outposts:region:123456789012:outpost/op-01ac5d28a6a232904/bucket/example-outposts-bucket
```

Ejemplos de política de bucket

Las políticas de bucket de S3 en Outposts le permiten proteger el acceso a los objetos de sus buckets de S3 en Outposts, de modo que solo los usuarios con los permisos adecuados puedan acceder a ellos. Incluso puede impedir que los usuarios autenticados que no dispongan de los permisos adecuados accedan a los recursos de S3 en Outposts.

En esta sección se presentan ejemplos de casos de uso típicos de políticas de bucket de S3 en Outposts. Para probar estas políticas, sustituya *`user input placeholders`* por su propia información (como el nombre del bucket).

Para conceder o denegar permisos a un conjunto de objetos, puede usar caracteres comodín (`*`) en nombres de recurso de Amazon (ARN) y otros valores. Por ejemplo, puede controlar el acceso a grupos de objetos que empiezan por un prefijo o terminar con una extensión dada, como `.html`.

Para obtener más información sobre el lenguaje de la política de AWS Identity and Access Management (IAM), consulte Configuración de IAM con S3 en Outposts.

> **Note**
>
> Si utiliza la consola de Amazon S3 para probar los permisos de s3outposts, debe conceder permisos adicionales requeridos por la consola como `s3outposts:createendpoint` o `s3outposts:listendpoints`, entre otros.

Recursos adicionales para crear políticas de bucket

- Para obtener una lista de las acciones, los recursos y las claves de condición de la política de IAM que puede utilizar al crear una política de bucket de S3 en Outposts, consulte Actions, resources, and condition keys for Amazon S3 on Outposts.
- Para obtener información sobre cómo crear una política de S3 en Outposts, consulte Adición o edición de una política de bucket para un bucket de Amazon S3 en Outposts.

Temas

- Gestión del acceso a un bucket de Amazon S3 en Outposts en función de determinadas direcciones IP

Gestión del acceso a un bucket de Amazon S3 en Outposts en función de determinadas direcciones IP

Una política de bucket es una política AWS Identity and Access Management basada en recursos (IAM) que puede utilizar para conceder permisos de acceso al bucket y a los objetos que contiene. Solo el propietario del bucket puede asociar una política a un bucket. Los permisos asociados a un bucket se aplican a todos los objetos del bucket que son propiedad de la cuenta de propietario del bucket. Las políticas de bucket tienen un límite de tamaño de 20 KB. Para obtener más información, consulte Política de bucket.

Restringir el acceso a direcciones IP específicas

En el siguiente ejemplo se impide que los usuarios realicen operaciones de S3 en Outposts en objetos en los buckets especificados, a menos que la solicitud se origine en el rango de direcciones IP especificado.

> **Note**
>
> Al restringir el acceso a una dirección IP concreta, asegúrese de especificar también qué puntos de conexión de VPC, direcciones IP de origen de VPC o direcciones IP externas pueden acceder al bucket de S3 en Outposts. De lo contrario, podría perder el acceso al bucket si su política deniega a todos los usuarios realizar cualquier operación de s3outposts en los objetos de su bucket de S3 en Outposts sin contar con los permisos adecuados.

La instrucción `Condition` de esta política identifica *`192.0.2.0/24`* como el rango de direcciones IP permitidas del IP versión 4 (IPv4).

El bloque `Condition` utiliza la condición `NotIpAddress` y la clave de condición `aws:SourceIp`, que es una clave de condición general de AWS. La clave de condición `aws:SourceIp` solo puede utilizarse para rangos de direcciones IP públicas. Para obtener más información acerca de estas claves de condición, consulte Actions, resources, and condition keys for S3 on Outposts. Los valores de IPv4 `aws:SourceIp` utilizan la notación CIDR estándar. Para obtener más información, consulte Referencia de los elementos de las políticas de JSON de IAM en la Guía del usuario de IAM.

> **Warning**
>
> Antes de utilizar esta política de S3 en Outposts, reemplace el rango de direcciones IP *`192.0.2.0/24`* de este ejemplo por un valor adecuado para su caso de uso. De lo contrario, ya no podrá acceder a su 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"
                }
            }
        }
    ]
}
```

Permitir direcciones IPv4 e IPv6

Cuando empiece a usar direcciones IPv6, le recomendamos que actualice todas las políticas de la organización con los rangos de direcciones IPv6 además de los rangos de direcciones IPv4 existentes. De este modo se asegurará de que las políticas sigan funcionando durante la transición a IPv6.

En el siguiente ejemplo de política de bucket de S3 en Outposts se muestra cómo combinar los rangos de dirección IPv4 e IPv6 para incluir todas las direcciones IP válidas de la organización. La política de ejemplo permite el acceso a las direcciones IP de ejemplo *192.0.2.1* y *2001:DB8:1234:5678::1*, y deniega el acceso a las direcciones *203.0.113.1* y *2001:DB8:1234:5678:ABCD::1*.

La clave de condición `aws:SourceIp` solo puede utilizarse para rangos de direcciones IP públicas. Los valores de IPv6 para `aws:SourceIp` deben estar en formato CIDR estándar. Para IPv6, aceptamos el uso de `::` para representar un rango de 0, (por ejemplo, `2001:DB8:1234:5678::/64`). Para obtener más información, consulte Operadores de condición de dirección IP en la Guía del usuario de IAM.

> **Warning**
>
> Antes de utilizar esta política de S3 en Outposts, sustituya los intervalos de direcciones IP del ejemplo por valores adecuados para su caso de uso. De lo contrario, puede perder la capacidad de acceder a su bucket.

```
{
    "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"
                    ]
                }
            }
        }
    ]
}
```

Obtención de una lista de buckets de Amazon S3 en Outposts

Con Amazon S3 en Outposts, puede crear buckets de S3 en Outposts de AWS y almacenar y recuperar fácilmente objetos en las instalaciones para las aplicaciones que requieren acceso local a los datos, procesamiento local de los datos y residencia de los datos. S3 en Outposts proporciona una nueva clase de almacenamiento, S3 Outposts (0UTP0STS), que utiliza las API de Amazon S3 y está diseñada para almacenar datos de manera duradera y redundante en múltiples dispositivos y servidores de AWS Outposts. Usted se comunica con su bucket de Outpost mediante un punto de acceso y una conexión de punto de conexión a través de una nube privada virtual (VPC). Puede usar las mismas API y características en los buckets de Outposts que en buckets de Amazon S3, como políticas de acceso, cifrado y etiquetado. Puede utilizar S3 en Outposts a través de la AWS

Management Console, AWS Command Line Interface (AWS CLI), AWS SDK o la API de REST. Para obtener más información, consulte ¿Qué es Amazon S3 en Outposts?.

Para obtener más información acerca de los buckets de S3 en Outposts, consulte Trabajo con buckets de S3 en Outposts.

En los siguientes ejemplos, se muestra cómo devolver una lista de los buckets de S3 en Outposts con AWS Management Console, AWS CLI y AWS SDK for Java.

Uso de la consola de S3

1. Abra la consola de Amazon S3 en https://console.aws.amazon.com/s3.
2. En el panel de navegación de la izquierda, elija Outposts buckets (Buckets de Outposts).
3. En Outposts buckets (Buckets de Outposts), revise la lista de buckets de S3 en Outposts.

Mediante AWS CLI

En el siguiente ejemplo de AWS CLI se obtiene una lista de buckets de un Outpost. Para usar este comando, sustituya *user input placeholder* por su propia información. Para obtener más información acerca de este comando, consulte list-regional-buckets en la Referencia de AWS CLI.

```
aws s3control list-regional-buckets --account-id 123456789012 --outpost-
id op-01ac5d28a6a232904
```

Uso de AWS SDK para Java

En el siguiente ejemplo del SDK para Java, se obtiene una lista de buckets de un Outpost. Para obtener más información, consulte ListRegionalBuckets en la Referencia de la API de Amazon Simple Storage Service.

```
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());

}
```

Obtención de un bucket de S3 en Outposts mediante la AWS CLI y el SDK para Java

Con Amazon S3 en Outposts, puede crear buckets de S3 en Outposts de AWS y almacenar y recuperar fácilmente objetos en las instalaciones para las aplicaciones que requieren acceso local a los datos, procesamiento local de los datos y residencia de los datos. S3 en Outposts proporciona una nueva clase de almacenamiento, S3 Outposts (OUTPOSTS), que utiliza las API de Amazon S3 y está diseñada para almacenar datos de manera duradera y redundante en múltiples dispositivos y servidores de AWS Outposts. Usted se comunica con su bucket de Outpost mediante un punto de acceso y una conexión de punto de conexión a través de una nube privada virtual (VPC). Puede usar las mismas API y características en los buckets de Outposts que en buckets de Amazon S3, como políticas de acceso, cifrado y etiquetado. Puede utilizar S3 en Outposts a través de la AWS Management Console, AWS Command Line Interface (AWS CLI), AWS SDK o la API de REST. Para obtener más información, consulte ¿Qué es Amazon S3 en Outposts?

En los siguientes ejemplos, se muestra cómo obtener un bucket de S3 en Outposts con AWS CLI y AWS SDK for Java.

> **Note**
>
> Al trabajar con Amazon S3 en Outposts a través de los SDK de AWS CLI o AWS, se proporciona el ARN del punto de acceso para Outpost en lugar del nombre del bucket. El ARN del punto de acceso adopta la siguiente forma, donde *region* es el código de Región de AWS de la región en la que está destinado el Outpost:
> `arn:aws:s3-outposts:region:123456789012:outpost/op-01ac5d28a6a232904/accesspoint/example-outposts-access-point`
> Para obtener más información acerca de S3 en Outposts, consulte ARN de recursos para S3 en Outposts.

Mediante AWS CLI

El siguiente ejemplo de S3 en Outposts obtiene un bucket con la AWS CLI. Para usar este comando, sustituya *user input placeholder* por su propia información. Para obtener más información acerca de este comando, consulte get-bucket en la Referencia de AWS CLI.

```
aws s3control get-bucket --account-id 123456789012 --bucket "arn:aws:s3-
outposts:region:123456789012:outpost/op-01ac5d28a6a232904/bucket/example-outposts-
bucket"
```

Uso de AWS SDK para Java

El siguiente ejemplo de S3 en Outposts obtiene un bucket con el SDK para Java. Para obtener más información, consulte GetBucket en la Referencia de la API de Amazon Simple Storage Service.

```
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());

}
```

Eliminación del bucket de Amazon S3 en Outposts

Con Amazon S3 en Outposts, puede crear buckets de S3 en Outposts de AWS y almacenar y recuperar fácilmente objetos en las instalaciones para las aplicaciones que requieren acceso local a los datos, procesamiento local de los datos y residencia de los datos. S3 en Outposts proporciona una nueva clase de almacenamiento, S3 Outposts (0UTP0STS), que utiliza las API de Amazon S3 y está diseñada para almacenar datos de manera duradera y redundante en múltiples dispositivos y servidores de AWS Outposts. Usted se comunica con su bucket de Outpost mediante un punto de acceso y una conexión de punto de conexión a través de una nube privada virtual (VPC). Puede usar las mismas API y características en los buckets de Outposts que en buckets de Amazon S3, como políticas de acceso, cifrado y etiquetado. Puede utilizar S3 en Outposts a través de la AWS

Management Console, AWS Command Line Interface (AWS CLI), AWS SDK o la API de REST. Para obtener más información, consulte ¿Qué es Amazon S3 en Outposts?.

Para obtener más información acerca de los buckets de S3 en Outposts, consulte Trabajo con buckets de S3 en Outposts.

La Cuenta de AWS que crea el bucket es su propietaria y la única que puede eliminarlo.

> **Note**
>
> - Los buckets de Outposts deben estar vacíos antes de que puedan eliminarse.
>
> La consola de Amazon S3 no admite acciones de objetos S3 en Outposts. Para eliminar objetos de bucket de S3 en Outposts, debe utilizar la API REST, AWS CLI o los SDK de AWS.
> - Antes de eliminar un bucket de Outposts, debe eliminar los puntos de acceso de Outposts del bucket. Para obtener más información, consulte Eliminar un punto de acceso .
> - No se puede recuperar un bucket después de que se haya eliminado.

En los siguientes ejemplos, se muestra cómo eliminar un bucket de S3 en Outposts con la AWS Management Console y AWS Command Line Interface (AWS CLI).

Uso de la consola de S3

1. Inicie sesión en la AWS Management Console y abra la consola de Amazon S3 en https://console.aws.amazon.com/s3/.
2. En el panel de navegación de la izquierda, elija Outposts buckets (Buckets de Outposts).
3. Elija el bucket que desea eliminar y elija Delete (Eliminar).
4. Confirme la eliminación.

Mediante AWS CLI

En el siguiente ejemplo, se elimina un bucket de S3 en Outposts (`s3-outposts:DeleteBucket`) con la AWS CLI. Para ejecutar este comando, sustituya los *`user input placeholders`* con su propia información.

```
aws s3control delete-bucket --account-id 123456789012 --bucket arn:aws:s3-
outposts:region:123456789012:outpost/op-01ac5d28a6a232904/bucket/example-outposts-
bucket
```

Trabajo con puntos de acceso de Amazon S3 en Outposts

Para acceder a su bucket de Amazon S3 en Outposts, debe crear y configurar un punto de acceso.

Los puntos de acceso simplifican la administración del acceso a los datos a escala para los conjuntos de datos compartidos en Amazon S3. Los puntos de acceso son puntos de enlace de red con nombre y asociados a los buckets que se pueden utilizar para realizar operaciones con objetos de Amazon S3, como `GetObject` y `PutObject`. Con S3 en Outposts, debe utilizar puntos de acceso para acceder a cualquier objeto de un bucket de Outposts. Los puntos de acceso solo admiten el direccionamiento de tipo de host virtual.

> **Note**
>
> La Cuenta de AWS que crea el bucket de Outposts es su propietaria y la única que puede asignarle puntos de acceso.

En las secciones siguientes, se describe cómo crear y administrar los puntos de acceso de buckets de S3 en Outposts.

Temas

Creación de un punto de acceso de S3 en Outposts

Para acceder a su bucket de Amazon S3 en Outposts, debe crear y configurar un punto de acceso.

Los puntos de acceso simplifican la administración del acceso a los datos a escala para los conjuntos de datos compartidos en Amazon S3. Los puntos de acceso son puntos de enlace de red con nombre y asociados a los buckets que se pueden utilizar para realizar operaciones con objetos de Amazon S3, como `GetObject` y `PutObject`. Con S3 en Outposts, debe utilizar puntos de acceso para acceder a cualquier objeto de un bucket de Outposts. Los puntos de acceso solo admiten el direccionamiento de tipo de host virtual.

En los siguientes ejemplos se muestra cómo crear un punto de acceso de S3 en Outposts mediante la AWS Management Console, AWS Command Line Interface (AWS CLI) y AWS SDK for Java.

> **Note**
>
> La Cuenta de AWS que crea el bucket de Outposts es su propietaria y la única que puede asignarle puntos de acceso.

Uso de la consola de S3

1. Abra la consola de Amazon S3 en https://console.aws.amazon.com/s3.
2. En el panel de navegación de la izquierda, elija Outposts buckets (Buckets de Outposts).
3. Seleccione el bucket de Outposts para el que desea crear un punto de acceso de Outposts.
4. Seleccione la pestaña Puntos de acceso de Outposts.
5. En la sección Outposts access points (Puntos de acceso de Outposts), elija Create Outposts access point (Crear punto de acceso de Outposts).
6. En la sección Outposts access point settings (Configuración del punto de acceso de Outposts), ingrese un nombre para el punto de acceso y elija la nube virtual privada (VPC) para el punto de acceso.
7. Si desea agregar una política para su punto de acceso, puede hacerlo ingresando en la sección Outposts access point policy (Política de punto de acceso de Outposts).

   Para obtener más información, consulte Configuración de IAM con S3 en Outposts .

Mediante AWS CLI

Example

En el siguiente ejemplo de la AWS CLI, se crea un punto de acceso para un bucket de Outposts. Para ejecutar este comando, sustituya los *`user input placeholders`* con su propia información.

```
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
```

Uso de AWS SDK para Java

Example

En el siguiente ejemplo del SDK para Java, se crea un punto de acceso para un bucket de Outposts. Para utilizar este ejemplo, reemplace los *user input placeholders* con su propia información.

```
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();

}
```

Uso de un alias de estilo de bucket para su punto de acceso de bucket de S3 en Outposts

Con S3 en Outposts, debe utilizar puntos de acceso para acceder a cualquier objeto de un bucket de Outposts. Cada vez que se crea un punto de acceso para un bucket, S3 en Outposts genera de forma automática un alias de punto de acceso. Puede utilizar este alias de punto de acceso en lugar de un ARN de punto de acceso para cualquier operación del plano de datos. Por ejemplo, puede usar un alias de punto de acceso para realizar operaciones a nivel de objeto, como PUT, GET, LIST y más. Para obtener una lista de las operaciones, consulte Operaciones de la API de Amazon S3 para administrar objetos.

Los siguientes ejemplos muestran un ARN y un alias de punto de acceso para un punto de acceso llamado *my-access-point*.

- ARN del punto de acceso: `arn:aws:s3-outposts:`*region*`:`*123456789012*`:outpost/`*op-01ac5d28a6a232904*`/accesspoint/`*my-access-point*
- Alias de punto de acceso: *my-access-po*-*o01ac5d28a6a232904e8xz5w8ijx1qzlbp3i3kuse10*`--op-s3`

Para obtener más información acerca de los ARN, consulte Nombres de recurso de Amazon (ARN) en la Referencia general de AWS.

Para obtener más información acerca de los alias de punto de acceso, consulte los siguientes temas.

Temas

- Alias de punto de acceso
- Uso de un alias de punto de acceso en una operación de objeto de S3 en Outposts
- Limitaciones

Alias de punto de acceso

Se crea un alias de punto de acceso en el mismo espacio de nombres que un bucket de Outposts. Al crear un punto de acceso, S3 en Outposts genera de forma automática un alias de punto de acceso que no se puede modificar. Un alias de punto de acceso cumple con todos los requisitos de un nombre de bucket válido de S3 en Outposts y consta de las siguientes partes:

access point name prefix-*metadata*`--op-s3`

> **Note**
>
> El sufijo `--op-s3` está reservado para los alias de punto de acceso, por lo que se recomienda no utilizarlo para los nombres de punto de acceso o bucket. Para obtener más información acerca de las reglas de nomenclatura del bucket de S3 en Outposts, consulte Trabajo con buckets de S3 en Outposts.

Búsqueda del alias de punto de acceso

En los siguientes ejemplos se muestra cómo encontrar un alias de punto de acceso utilizando la consola de Amazon S3 y la AWS CLI.

Example : Buscar y copiar un alias de punto de acceso en la consola de Amazon S3

Después de crear un punto de acceso en la consola, puede obtener el alias del punto de acceso en la columna Access Point alias (Alias de puntos de acceso) de la lista Access Points (Puntos de acceso).

Para copiar un alias de punto de acceso

1. Abra la consola de Amazon S3 en https://console.aws.amazon.com/s3.
2. En el panel de navegación de la izquierda, elija Outposts access points (Puntos de acceso de Outposts).
3. Para copiar el alias de punto de acceso, realice una de las siguientes acciones:
   - En la lista Access Points (Puntos de acceso), seleccione el botón de opción situado junto al nombre del punto de acceso y, a continuación, elija Copy Access Point alias (Copiar alias de punto de acceso).
   - Seleccione el nombre del punto de acceso. A continuación, en Outposts access point overview (Descripción general del punto de acceso de Outposts), copie el alias del punto de acceso.

Example : Crear un punto de acceso utilizando la AWS CLI y buscar el alias del punto de acceso en la respuesta

El siguiente ejemplo de la AWS CLI del comando `create-access-point` crea el punto de acceso y devuelve el alias de punto de acceso generado automáticamente. Para ejecutar este comando, sustituya los *`user input placeholders`* con su propia información.

```
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 : Obtener un alias de punto de acceso utilizando la AWS CLI

El siguiente ejemplo de AWS CLI del comando `get-access-point` devuelve información sobre el punto de acceso especificado. Esta información incluye el alias de punto de acceso. Para ejecutar este comando, sustituya los *`user input placeholders`* con su propia información.

```
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 : Enumerar los puntos de acceso para encontrar un alias de punto de acceso mediante la AWS CLI

El siguiente ejemplo de AWS CLI del comando `list-access-points` enumera información sobre el punto de acceso especificado. Esta información incluye el alias de punto de acceso. Para ejecutar este comando, sustituya los *`user input placeholders`* con su propia información.

```
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"
        }
    ]
}
```

Uso de un alias de punto de acceso en una operación de objeto de S3 en Outposts

Al adoptar puntos de acceso, puede utilizar alias de puntos de acceso sin tener que hacer cambios exhaustivos en el código.

En este ejemplo de AWS CLI se muestra una operación `get-object` para un bucket de S3 en Outposts. En este ejemplo, se utiliza el alias del punto de acceso como el valor de `--bucket`, en lugar del ARN completo del punto de acceso.

```
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": {}
}
```

Limitaciones

- Los clientes no pueden configurar los alias.
- Los alias no se pueden eliminar, modificar ni dehabillitar en un punto de acceso.

- No puede utilizar un alias de punto de acceso para operaciones de plano de control de S3 en Outposts. Para ver la lista de operaciones del plano de control de S3 en Outposts, consulte Operaciones de la API de Amazon S3 Control para administrar buckets.
- Los alias no se pueden usar en las políticas de AWS Identity and Access Management (IAM).

Visualización de información acerca de una configuración de punto de acceso

Los puntos de acceso simplifican la administración del acceso a los datos a escala para los conjuntos de datos compartidos en Amazon S3. Los puntos de acceso son puntos de enlace de red con nombre y asociados a los buckets que se pueden utilizar para realizar operaciones con objetos de Amazon S3, como `GetObject` y `PutObject`. Con S3 en Outposts, debe utilizar puntos de acceso para acceder a cualquier objeto de un bucket de Outposts. Los puntos de acceso solo admiten el direccionamiento de tipo de host virtual.

Los siguientes temas muestran cómo devolver información de configuración de un punto de acceso de S3 en Outposts mediante la AWS Management Console, AWS Command Line Interface (AWS CLI) y AWS SDK for Java.

Uso de la consola de S3

1. Abra la consola de Amazon S3 en https://console.aws.amazon.com/s3.
2. En el panel de navegación de la izquierda, elija Outposts access points (Puntos de acceso de Outposts).
3. Elija el punto de acceso de Outposts para el que desea ver los detalles de configuración.
4. En Outposts access point overview (Resumen del punto de acceso de Outposts), revise los detalles de configuración del punto de acceso.

Mediante AWS CLI

En el siguiente ejemplo de la AWS CLI, se obtiene un punto de acceso para un bucket de Outposts. Sustituya los *`user input placeholders`* con su propia información.

```
aws s3control get-access-point --account-id 123456789012 --name arn:aws:s3-
outposts:region:123456789012:outpost/op-01ac5d28a6a232904/accesspoint/example-outposts-
access-point
```

Uso de AWS SDK para Java

En el siguiente ejemplo del SDK para Java, se obtiene un punto de acceso para un bucket de 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());

}
```

Visualización de una lista de puntos de acceso de Amazon S3 en Outposts

Los puntos de acceso simplifican la administración del acceso a los datos a escala para los conjuntos de datos compartidos en Amazon S3. Los puntos de acceso son puntos de enlace de red con nombre y asociados a los buckets que se pueden utilizar para realizar operaciones con objetos de Amazon S3, como `GetObject` y `PutObject`. Con S3 en Outposts, debe utilizar puntos de acceso para acceder a cualquier objeto de un bucket de Outposts. Los puntos de acceso solo admiten el direccionamiento de tipo de host virtual.

En los siguientes temas, se muestra cómo devolver una lista de los puntos de acceso de S3 en Outposts mediante la AWS Management Console, AWS Command Line Interface (AWS CLI) y AWS SDK for Java.

Uso de la consola de S3

1. Abra la consola de Amazon S3 en https://console.aws.amazon.com/s3.
2. En el panel de navegación de la izquierda, elija Outposts access points (Puntos de acceso de Outposts).
3. En Outposts access points (Puntos de acceso de Outposts), revise la lista de puntos de acceso de S3 en Outposts.

Mediante AWS CLI

En el siguiente ejemplo de la AWS CLI, se enumeran los puntos de acceso para un bucket de Outposts. Para ejecutar este comando, sustituya los *user input placeholders* con su propia información.

```
aws s3control list-access-points --account-id 123456789012 --bucket arn:aws:s3-
outposts:region:123456789012:outpost/op-01ac5d28a6a232904/bucket/example-outposts-
bucket
```

Uso de AWS SDK para Java

En el siguiente ejemplo del SDK para Java, se enumeran los puntos de acceso para un bucket de 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());

}
```

Eliminar un punto de acceso

Los puntos de acceso simplifican la administración del acceso a los datos a escala para los conjuntos de datos compartidos en Amazon S3. Los puntos de acceso son puntos de enlace de red con nombre y asociados a los buckets que se pueden utilizar para realizar operaciones con objetos de Amazon S3, como `GetObject` y `PutObject`. Con S3 en Outposts, debe utilizar puntos de acceso para acceder a cualquier objeto de un bucket de Outposts. Los puntos de acceso solo admiten el direccionamiento de tipo de host virtual.

En los siguientes ejemplos, se muestra cómo eliminar un punto de acceso mediante AWS Management Console y la AWS Command Line Interface (AWS CLI).

Uso de la consola de S3

1. Abra la consola de Amazon S3 en https://console.aws.amazon.com/s3.
2. En el panel de navegación de la izquierda, elija Outposts access points (Puntos de acceso de Outposts).
3. En la sección Outposts access points (Puntos de acceso de Outposts), elija el punto de acceso de Outposts que desea eliminar.
4. Elija Eliminar.
5. Confirme la eliminación.

Mediante AWS CLI

En el siguiente ejemplo de la AWS CLI, se elimina un punto de acceso de Outposts. Para ejecutar este comando, sustituya los *user input placeholders* con su propia información.

```
aws s3control delete-access-point --account-id 123456789012 --name arn:aws:s3-
outposts:region:123456789012:outpost/op-01ac5d28a6a232904/accesspoint/example-outposts-
access-point
```

Adición o edición de una política de punto de acceso

Cada punto de acceso tiene permisos y controles de red distintos que Amazon S3 en Outposts se aplica a cualquier solicitud que se realice a través de ese punto de acceso. Cada punto de acceso aplica una política de punto de acceso personalizada que funciona en conjunción con la política de bucket asociada al bucket subyacente. Para obtener más información, consulte Puntos de acceso .

En los siguientes temas, se muestra cómo agregar o editar la política de punto de acceso de su punto de acceso de S3 en Outposts mediante la AWS Management Console, AWS Command Line Interface (AWS CLI) y AWS SDK for Java.

Uso de la consola de S3

1. Abra la consola de Amazon S3 en https://console.aws.amazon.com/s3.
2. En el panel de navegación de la izquierda, elija Outposts buckets (buckets de Outposts).
3. Elija el bucket de Outposts para el que desea editar la política de punto de acceso.
4. Seleccione la pestaña Puntos de acceso de Outposts.

5. En la sección Outposts access points (Puntos de acceso de Outposts), seleccione el punto de acceso cuya política desea editar y elija Edit policy (Editar política).
6. Agregue o edite la política en la sección Outposts access point policy (política de puntos de acceso de Outposts) . Para obtener más información, consulte Configuración de IAM con S3 en Outposts .

Mediante AWS CLI

En el siguiente ejemplo de la AWS CLI, se aplica una política en un punto de acceso de Outposts.

1. Guarde la siguiente política de punto de acceso en un archivo JSON. En este ejemplo, el archivo se denomina `appolicy1.json`. Sustituya los *`user input placeholders`* con su propia información.

```
{
   "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. Envíe el archivo JSON como parte del comando de la CLI `put-access-point-policy`. Sustituya los *`user input placeholders`* con su propia información.

```
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
```

Uso de AWS SDK para Java

En el siguiente ejemplo del SDK para Java, se aplica una política en un punto de acceso de 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());

}
```

Visualización de una política de punto de acceso para un punto de acceso de S3 en Outposts

Cada punto de acceso tiene permisos y controles de red distintos que Amazon S3 en Outposts aplica a cualquier solicitud que se realice a través de ese punto de acceso. Cada punto de acceso aplica una política de punto de acceso personalizada que funciona en conjunción con la política de bucket asociada al bucket subyacente. Para obtener más información, consulte Puntos de acceso .

Para obtener más información acerca del uso de puntos de acceso en S3 en Outposts, consulte Trabajo con buckets de S3 en Outposts.

En los siguientes temas, se muestra cómo ver la política de punto de acceso de S3 en Outposts utilizando la AWS Management Console, AWS Command Line Interface (AWS CLI) y AWS SDK for Java.

Uso de la consola de S3

1. Abra la consola de Amazon S3 en https://console.aws.amazon.com/s3.
2. En el panel de navegación de la izquierda, elija Outposts access points (Puntos de acceso de Outposts).
3. Elija el punto de acceso de Outposts para el que desea ver la política.
4. En la página Permissions (Permisos), revise la política del punto de acceso de S3 en Outposts.
5. Para editar la política de punto de acceso, consulte Adición o edición de una política de punto de acceso.

Mediante AWS CLI

En el siguiente ejemplo de la AWS CLI, se obtiene una política para un punto de acceso de Outposts. Para ejecutar este comando, sustituya los *user input placeholders* con su propia información.

```
aws s3control get-access-point-policy --account-id 123456789012 --name arn:aws:s3-
outposts:region:123456789012:outpost/op-01ac5d28a6a232904/accesspoint/example-outposts-
access-point
```

Uso de AWS SDK para Java

En el siguiente ejemplo del SDK para Java, se obtiene una política para un punto de acceso de 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());

}
```

Trabajo con puntos de conexión de Amazon S3 en Outposts

Para dirigir solicitudes a un punto de acceso de Amazon S3 en Outposts, debe crear y configurar un punto de conexión de S3 en Outposts. Para crear un punto de conexión, necesita una conexión activa con el enlace de servicio a la región de origen de Outposts. Cada nube virtual privada (VPC) de su Outpost puede tener un punto de conexión asociado. Para obtener más información acerca de las cuotas de los puntos de conexión, consulte Requisitos de red de S3 en Outposts. Debe crear un punto de conexión para poder acceder a los buckets de Outposts y realizar operaciones de objetos. Para obtener más información, consulte Puntos de conexión.

Después de crear un punto de conexión, puede utilizar el campo Estado para comprender el estado del punto de conexión. Si Outposts está desconectado, devolverá un error CREATE_FAILED. Puede comprobar la conexión del enlace de servicio, eliminar el punto de conexión y volver a intentar la operación de creación cuando se haya reanudado la conexión. Para obtener una lista de códigos de error adicionales, consulte la siguiente tabla. Para obtener más información, consulte Puntos de conexión.

API	Estado	Código de error de motivo de error	Mensaje - Motivo del error
`CreateEndpoint`	Create_Failed	OutpostNotReachable	No se ha podido crear un punto de conexión porque la conexión del enlace de servicio a la región de origen de Outposts está inactiva. Compruebe la conexión, borre el punto de conexión e inténtelo de nuevo.
`CreateEndpoint`	Create_Failed	InternalError	No se ha podido crear el punto de conexión debido a un error interno. Elimine el punto de conexión y vuelva a crearlo.

API	Estado	Código de error de motivo de error	Mensaje - Motivo del error
`DeleteEndpoint`	Delete_Failed	OutpostNotReachable	No se ha podido eliminar un punto de conexión porque la conexión del enlace de servicio a la región de origen de Outposts está inactiva. Compruebe la conexión e inténtelo de nuevo.
`DeleteEndpoint`	Delete_Failed	InternalError	No se ha podido eliminar el punto de conexión debido a un error interno. Inténtelo de nuevo.

Para obtener más información acerca de trabajar con buckets en S3 en Outposts, consulte Trabajo con buckets de S3 en Outposts.

En las secciones siguientes, se describe cómo crear y administrar puntos de conexión para S3 en Outposts.

Temas

- Creación de un punto de conexión en un Outpost
- Obtención de una lista de puntos de conexión de Amazon S3 en Outposts
- Eliminación de un punto de conexión de Amazon S3 en Outposts

Creación de un punto de conexión en un Outpost

Para dirigir solicitudes a un punto de acceso de Amazon S3 en Outposts, debe crear y configurar un punto de conexión de S3 en Outposts. Para crear un punto de conexión, necesita una conexión activa con el enlace de servicio a la región de origen de Outposts. Cada nube virtual privada (VPC) de su Outpost puede tener un punto de conexión asociado. Para obtener más información acerca de las cuotas de los puntos de conexión, consulte Requisitos de red de S3 en Outposts. Debe crear un punto de conexión para poder acceder a los buckets de Outposts y realizar operaciones de objetos. Para obtener más información, consulte Puntos de conexión.

Permisos

Para obtener más información sobre los permisos necesarios para crear un punto de conexión, consulte Permisos para los puntos de conexión de S3 en Outposts.

Al crear un punto de conexión, S3 en Outposts también crea un rol vinculado a un servicio en la Cuenta de AWS. Para obtener más información, consulte Uso de roles vinculados a servicios para Amazon S3 en Outposts.

En los siguientes ejemplos, se muestra cómo crear un punto de conexión de S3 en Outposts con AWS Management Console, AWS Command Line Interface (AWS CLI) y AWS SDK for Java.

Uso de la consola de S3

1. Inicie sesión en la AWS Management Console y abra la consola de Amazon S3 en https://console.aws.amazon.com/s3/.
2. En el panel de navegación de la izquierda, elija Outposts access points (Puntos de acceso de Outposts).
3. Seleccione la pestaña Outposts endpoints (Puntos de acceso de Outposts).
4. Elija Create Outposts endpoint (Crear punto de conexión de Outposts).
5. En Outpost, elija el Outpost en el que crear este punto de conexión.
6. En VPC, elija una VPC que aún no tenga punto de conexión y que también cumpla las reglas de los puntos de conexión de Outposts.

   Una nube virtual privada (VPC) le permite lanzar recursos de AWS en una red virtual que defina. Dicha red virtual es prácticamente idéntica a las redes tradicionales que se utilizarían en sus propios centros de datos, con los beneficios que supone utilizar la infraestructura escalable de AWS.

   Si no tiene una VPC, elija Create VPC (Crear VPC). Para obtener más información, consulte Crear puntos de acceso restringidos a una nube privada virtual.
7. Elija Create Outposts endpoint (Crear punto de conexión de Outposts).

Utilización de la AWS CLI

Example

En el siguiente ejemplo de la AWS CLI, se crea un punto de conexión para un Outpost con el tipo de acceso a recursos de la VPC. La VPC se obtiene de la subred. Para ejecutar este comando, sustituya los *user input placeholders* con su propia información.

```
aws s3outposts create-endpoint --outpost-id op-01ac5d28a6a232904 --subnet-id
 subnet-8c7a57c5 --security-group-id sg-ab19e0d1
```

En el siguiente ejemplo de la AWS CLI, se crea un punto de conexión para un Outpost con el tipo de acceso de grupo de direcciones IP propiedad del cliente (grupo de CoIP). Para ejecutar este comando, sustituya los *user input placeholders* con su propia información.

```
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
```

Uso de AWS SDK para Java

Example

En el siguiente ejemplo del SDK para Java, se crea un punto de conexión para un Outpost. Para utilizar este ejemplo, reemplace los *user input placeholders* con su propia información.

```
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());
}
```

Obtención de una lista de puntos de conexión de Amazon S3 en Outposts

Para dirigir solicitudes a un punto de acceso de Amazon S3 en Outposts, debe crear y configurar un punto de conexión de S3 en Outposts. Para crear un punto de conexión, necesita una conexión activa con el enlace de servicio a la región de origen de Outposts. Cada nube virtual privada (VPC) de su Outpost puede tener un punto de conexión asociado. Para obtener más información acerca de las cuotas de los puntos de conexión, consulte Requisitos de red de S3 en Outposts. Debe crear un punto de conexión para poder acceder a los buckets de Outposts y realizar operaciones de objetos. Para obtener más información, consulte Puntos de conexión.

En los siguientes ejemplos, se muestra cómo devolver una lista de los puntos de conexión de S3 en Outposts con AWS Management Console, AWS Command Line Interface (AWS CLI) y AWS SDK for Java.

Uso de la consola de S3

1. Abra la consola de Amazon S3 en https://console.aws.amazon.com/s3.
2. En el panel de navegación de la izquierda, elija Outposts access points (Puntos de acceso de Outposts).
3. En la página Outposts access points (Puntos de acceso de Outposts), seleccione la pestaña Outposts endpoints (Puntos de conexión de Outposts).
4. En Outposts endpoints (Puntos de conexión de Outposts), puede ver una lista de sus puntos de conexión de S3 en Outposts.

Utilización de la AWS CLI

En el siguiente ejemplo de AWS CLI, se muestran los puntos de conexión para recursos de AWS Outposts asociados a la cuenta. Para obtener más información acerca de este comando, consulte list-endpoints en la Referencia de AWS CLI.

```
aws s3outposts list-endpoints
```

Uso de AWS SDK para Java

En el siguiente ejemplo del SDK para Java, se enumeran los puntos de enlace para un Outpost. Para obtener más información, consulte ListEndpoints en la Referencia de la API de Amazon Simple Storage Service.

```
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);
}
```

Eliminación de un punto de conexión de Amazon S3 en Outposts

Para dirigir solicitudes a un punto de acceso de Amazon S3 en Outposts, debe crear y configurar un punto de conexión de S3 en Outposts. Para crear un punto de conexión, necesita una conexión activa con el enlace de servicio a la región de origen de Outposts. Cada nube virtual privada (VPC) de su Outpost puede tener un punto de conexión asociado. Para obtener más información acerca de las cuotas de los puntos de conexión, consulte Requisitos de red de S3 en Outposts. Debe crear un punto de conexión para poder acceder a los buckets de Outposts y realizar operaciones de objetos. Para obtener más información, consulte Puntos de conexión.

En los siguientes ejemplos, se muestra cómo eliminar los puntos de conexión de S3 en Outposts con la AWS Management Console, AWS Command Line Interface (AWS CLI) y AWS SDK for Java.

Uso de la consola de S3

1. Abra la consola de Amazon S3 en https://console.aws.amazon.com/s3.
2. En el panel de navegación de la izquierda, elija Outposts access points (Puntos de acceso de Outposts).
3. En la página Outposts access points (Puntos de acceso de Outposts), seleccione la pestaña Outposts endpoints (Puntos de conexión de Outposts).
4. En Outposts endpoints (Puntos de conexión de Outposts), seleccione el punto de conexión que desea eliminar y elija Delete (Eliminar).

Utilización de la AWS CLI

En el siguiente ejemplo de la AWS CLI, se elimina un punto de enlace para un Outpost. Para ejecutar este comando, sustituya los *user input placeholders* con su propia información.

```
aws s3outposts delete-endpoint --endpoint-id example-endpoint-id --outpost-
id op-01ac5d28a6a232904
```

Uso de AWS SDK para Java

En el siguiente ejemplo del SDK para Java, se elimina un punto de enlace para un Outpost. Para utilizar este ejemplo, reemplace los *user input placeholders* con su propia información.

```
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.");
}
```

Trabajo con objetos de S3 en Outposts

Con Amazon S3 en Outposts, puede crear buckets de S3 en Outposts de AWS y almacenar y recuperar fácilmente objetos en las instalaciones para las aplicaciones que requieren acceso local a los datos, procesamiento local de los datos y residencia de los datos. S3 en Outposts proporciona una nueva clase de almacenamiento, S3 Outposts (OUTPOSTS), que utiliza las API de Amazon S3 y está diseñada para almacenar datos de manera duradera y redundante en múltiples dispositivos

y servidores de AWS Outposts. Usted se comunica con su bucket de Outpost mediante un punto de acceso y una conexión de punto de conexión a través de una nube privada virtual (VPC). Puede usar las mismas API y características en los buckets de Outposts que en buckets de Amazon S3, como políticas de acceso, cifrado y etiquetado. Puede utilizar S3 en Outposts a través de la AWS Management Console, AWS Command Line Interface (AWS CLI), AWS SDK o la API de REST.

Los objetos son las entidades fundamentales almacenadas en Amazon S3 en Outposts. Cada objeto está almacenado en un bucket. Debe utilizar puntos de acceso para acceder a cualquier objeto de un bucket de Outpost. Cuando especifica el bucket para las operaciones de objetos, se utiliza el Nombre de recurso de Amazon (ARN) del punto de acceso o el alias del punto de acceso. Para obtener más información acerca de los alias de punto de acceso, consulte Uso de un alias de estilo de bucket para su punto de acceso de bucket de S3 en Outposts.

En el siguiente ejemplo se muestra el formato ARN para los puntos de acceso de S3 en Outposts, que incluye el código Región de AWS de la Región a la que pertenece el Outpost, el ID de Cuenta de AWS, el ID de Outposts y el nombre del punto de acceso:

```
arn:aws:s3-outposts:region:account-id:outpost/outpost-id/accesspoint/accesspoint-name
```

Para obtener más información acerca de S3 en Outposts, consulte ARN de recursos para S3 en Outposts.

Los ARN de objeto utilizan el siguiente formato, que incluye la Región de AWS a la que está destinada Outposts, el ID de Cuenta de AWS, el ID de Outpost, el nombre del bucket y la clave de objeto:

```
arn:aws:s3-outposts:us-west-2:123456789012:outpost/ op-01ac5d28a6a232904/bucket/amzn-s3-demo-bucket1/object/myobject
```

Con Amazon S3 en Outposts, los datos de objeto siempre se almacenan en el Outpost. Cuando AWS instala un bastidor de Outpost, sus datos permanecen de manera local en su Outpost para cumplir los requisitos de residencia de datos. Sus objetos nunca salen de su Outpost y no están en una Región de AWS. Ya que la AWS Management Console está alojada dentro de la región, no puede usar la consola para cargar o administrar objetos en su Outpost. Sin embargo, puede utilizar la API de REST, AWS Command Line Interface (AWS CLI) y los SDK de AWS para cargar y administrar los objetos a través de los puntos de acceso.

Temas

- Carga de un objeto en un bucket de S3 en Outpost
- Copia de un objeto en un bucket de Amazon S3 en Outposts utilizando AWS SDK for Java
- Obtención de un objeto de un bucket de Amazon S3 en Outposts
- Obtención de listas de objetos en un bucket de Amazon S3 en Outposts
- Eliminación de objetos en buckets de Amazon S3 en Outposts
- Uso de HeadBucket para determinar si existe un bucket de S3 en Outposts y si tiene permisos de acceso
- Ejecución y administración de una carga multiparte con el SDK para Java
- Uso de URL prefirmadas para S3 en Outposts
- Amazon S3 en Outposts con Amazon EMR en Outposts local
- Almacenamiento en caché de autorización y autenticación

Carga de un objeto en un bucket de S3 en Outpost

Los objetos son las entidades fundamentales almacenadas en Amazon S3 en Outposts. Cada objeto está almacenado en un bucket. Debe utilizar puntos de acceso para acceder a cualquier objeto de un bucket de Outpost. Cuando especifica el bucket para las operaciones de objetos, se utiliza el Nombre de recurso de Amazon (ARN) del punto de acceso o el alias del punto de acceso. Para obtener más información acerca de los alias de punto de acceso, consulte Uso de un alias de estilo de bucket para su punto de acceso de bucket de S3 en Outposts.

En el siguiente ejemplo se muestra el formato ARN para los puntos de acceso de S3 en Outposts, que incluye el código Región de AWS de la Región a la que pertenece el Outpost, el ID de Cuenta de AWS, el ID de Outposts y el nombre del punto de acceso:

```
arn:aws:s3-outposts:region:account-id:outpost/outpost-id/accesspoint/accesspoint-name
```

Para obtener más información acerca de S3 en Outposts, consulte ARN de recursos para S3 en Outposts.

Con Amazon S3 en Outposts, los datos de objeto siempre se almacenan en el Outpost. Cuando AWS instala un bastidor de Outpost, sus datos permanecen de manera local en su Outpost para cumplir los requisitos de residencia de datos. Sus objetos nunca salen de su Outpost y no están en una Región de AWS. Ya que la AWS Management Console está alojada dentro de la región, no puede usar la consola para cargar o administrar objetos en su Outpost. Sin embargo, puede

utilizar la API de REST, AWS Command Line Interface (AWS CLI) y los SDK de AWS para cargar y administrar los objetos a través de los puntos de acceso.

Los siguientes ejemplos de AWS CLI y AWS SDK for Java muestran cómo cargar un objeto en un bucket de S3 en Outposts mediante un punto de acceso.

AWS CLI

Example

En el siguiente ejemplo, se aplica un objeto denominado `sample-object.xml` en un bucket de S3 en Outposts (`s3-outposts:PutObject`) mediante la AWS CLI. Para usar este comando, sustituya *`user input placeholder`* por su propia información. Para obtener más información acerca de este comando, consulte put-object en la Referencia de AWS CLI.

```
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

En el siguiente ejemplo, se aplica un objeto en un bucket de S3 en Outposts mediante el SDK para Java. Para utilizar este ejemplo, sustituya *`user input placeholder`* por su propia información. Para obtener más información, consulte Carga de objetos.

```
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();
        }
    }
}
```

Copia de un objeto en un bucket de Amazon S3 en Outposts utilizando AWS SDK for Java

Los objetos son las entidades fundamentales almacenadas en Amazon S3 en Outposts. Cada objeto está almacenado en un bucket. Debe utilizar puntos de acceso para acceder a cualquier objeto de un bucket de Outpost. Cuando especifica el bucket para las operaciones de objetos, se utiliza el Nombre de recurso de Amazon (ARN) del punto de acceso o el alias del punto de acceso. Para obtener más información acerca de los alias de punto de acceso, consulte Uso de un alias de estilo de bucket para su punto de acceso de bucket de S3 en Outposts.

En el siguiente ejemplo se muestra el formato ARN para los puntos de acceso de S3 en Outposts, que incluye el código Región de AWS de la Región a la que pertenece el Outpost, el ID de Cuenta de AWS, el ID de Outposts y el nombre del punto de acceso:

```
arn:aws:s3-outposts:region:account-id:outpost/outpost-id/accesspoint/accesspoint-name
```

Para obtener más información acerca de S3 en Outposts, consulte ARN de recursos para S3 en Outposts.

Con Amazon S3 en Outposts, los datos de objeto siempre se almacenan en el Outpost. Cuando AWS instala un bastidor de Outpost, sus datos permanecen de manera local en su Outpost para cumplir los requisitos de residencia de datos. Sus objetos nunca salen de su Outpost y no están en una Región de AWS. Ya que la AWS Management Console está alojada dentro de la región, no puede usar la consola para cargar o administrar objetos en su Outpost. Sin embargo, puede utilizar la API de REST, AWS Command Line Interface (AWS CLI) y los SDK de AWS para cargar y administrar los objetos a través de los puntos de acceso.

En los siguientes ejemplos, se muestra cómo obtener una lista de objetos de bucket de S3 en Outposts con AWS SDK for Java.

Uso de AWS SDK para Java

En el siguiente ejemplo de S3 en Outposts, se copia un objeto a un objeto nuevo en el mismo bucket con el SDK para Java. Para utilizar este ejemplo, reemplace los *user input placeholders* con su propia información.

```
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();
        }
    }
}
```

Obtención de un objeto de un bucket de Amazon S3 en Outposts

Los objetos son las entidades fundamentales almacenadas en Amazon S3 en Outposts. Cada objeto está almacenado en un bucket. Debe utilizar puntos de acceso para acceder a cualquier objeto de un bucket de Outpost. Cuando especifica el bucket para las operaciones de objetos, se utiliza el Nombre de recurso de Amazon (ARN) del punto de acceso o el alias del punto de acceso. Para obtener más información acerca de los alias de punto de acceso, consulte Uso de un alias de estilo de bucket para su punto de acceso de bucket de S3 en Outposts.

En el siguiente ejemplo se muestra el formato ARN para los puntos de acceso de S3 en Outposts, que incluye el código Región de AWS de la Región a la que pertenece el Outpost, el ID de Cuenta de AWS, el ID de Outposts y el nombre del punto de acceso:

```
arn:aws:s3-outposts:region:account-id:outpost/outpost-id/accesspoint/accesspoint-name
```

Para obtener más información acerca de S3 en Outposts, consulte ARN de recursos para S3 en Outposts.

Con Amazon S3 en Outposts, los datos de objeto siempre se almacenan en el Outpost. Cuando AWS instala un bastidor de Outpost, sus datos permanecen de manera local en su Outpost para cumplir los requisitos de residencia de datos. Sus objetos nunca salen de su Outpost y no están en una Región de AWS. Ya que la AWS Management Console está alojada dentro de la región, no puede usar la consola para cargar o administrar objetos en su Outpost. Sin embargo, puede utilizar la API de REST, AWS Command Line Interface (AWS CLI) y los SDK de AWS para cargar y administrar los objetos a través de los puntos de acceso.

Los siguientes ejemplos muestran cómo descargar (obtener) un objeto mediante AWS Command Line Interface (AWS CLI) y AWS SDK for Java.

Mediante AWS CLI

En el siguiente ejemplo, se obtiene un objeto denominado `sample-object.xml` de un bucket de S3 en Outposts (`s3-outposts:GetObject`) mediante la AWS CLI. Para usar este comando, sustituya *`user input placeholder`* por su propia información. Para obtener más información acerca de este comando, consulte get-object en la Referencia de AWS CLI.

```
aws s3api get-object --bucket arn:aws:s3-
outposts:region:123456789012:outpost/op-01ac5d28a6a232904/accesspoint/example-outposts-
access-point --key testkey sample-object.xml
```

Uso de AWS SDK para Java

En el siguiente ejemplo de S3 en Outposts, se obtiene un objeto mediante el SDK para Java. Para utilizar este ejemplo, sustituya *`user input placeholder`* por su propia información. Para obtener más información, consulte GetObject en la referencia de la API de 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();
    }
}
```

Obtención de listas de objetos en un bucket de Amazon S3 en Outposts

Los objetos son las entidades fundamentales almacenadas en Amazon S3 en Outposts. Cada objeto está almacenado en un bucket. Debe utilizar puntos de acceso para acceder a cualquier objeto de un bucket de Outpost. Cuando especifica el bucket para las operaciones de objetos, se utiliza el Nombre de recurso de Amazon (ARN) del punto de acceso o el alias del punto de acceso. Para obtener más información acerca de los alias de punto de acceso, consulte Uso de un alias de estilo de bucket para su punto de acceso de bucket de S3 en Outposts.

En el siguiente ejemplo se muestra el formato ARN para los puntos de acceso de S3 en Outposts, que incluye el código Región de AWS de la Región a la que pertenece el Outpost, el ID de Cuenta de AWS, el ID de Outposts y el nombre del punto de acceso:

```
arn:aws:s3-outposts:region:account-id:outpost/outpost-id/accesspoint/accesspoint-name
```

Para obtener más información acerca de S3 en Outposts, consulte ARN de recursos para S3 en Outposts.

> **Note**
>
> Con Amazon S3 en Outposts, los datos de objeto siempre se almacenan en el Outpost. Cuando AWS instala un bastidor de Outpost, sus datos permanecen de manera local en su Outpost para cumplir los requisitos de residencia de datos. Sus objetos nunca salen de su Outpost y no están en una Región de AWS. Ya que la AWS Management Console está alojada dentro de la región, no puede usar la consola para cargar o administrar objetos en su Outpost. Sin embargo, puede utilizar la API de REST, AWS Command Line Interface (AWS CLI) y los SDK de AWS para cargar y administrar los objetos a través de los puntos de acceso.

En los siguientes ejemplos, se muestra cómo obtener una lista de objetos de un bucket de S3 en Outposts mediante AWS CLI y AWS SDK for Java.

Mediante AWS CLI

En el siguiente ejemplo, se muestran los objetos en un bucket de S3 en Outposts (`s3-outposts:ListObjectsV2`) mediante AWS CLI. Para usar este comando, sustituya *`user input placeholder`* por su propia información. Para obtener más información acerca de este comando, consulte list-objects-v2 en la Referencia de AWS CLI.

```
aws s3api list-objects-v2 --bucket arn:aws:s3-outposts:region:123456789012:outpost/op-01ac5d28a6a232904/accesspoint/example-outposts-access-point
```

> **Note**
>
> Al utilizar esta acción con Amazon S3 en Outposts a través de SDK de AWS, proporciona el ARN del punto de acceso de Outposts en lugar del nombre del bucket, en la siguiente manera: `arn:aws:s3-outposts:region:123456789012:outpost/op-01ac5d28a6a232904/`

`accesspoint/`*`example-Outposts-Access-Point`*. Para obtener más información acerca de S3 en Outposts, consulte ARN de recursos para S3 en Outposts.

Uso de AWS SDK para Java

En el siguiente ejemplo de S3 en Outposts, se muestran objetos en un bucket mediante el SDK para Java. Para utilizar este ejemplo, sustituya *`user input placeholder`* por su propia información.

> **Important**
>
> En este ejemplo se utiliza ListObjectsV2, que es la revisión más reciente de la operación de la API `ListObjects`. Recomendamos usar esta operación de API revisada para el desarrollo de aplicaciones. Para garantizar la compatibilidad con versiones anteriores, Amazon S3 aún es compatible con la versión anterior de esta operación de 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();
        }
    }
}
```

Eliminación de objetos en buckets de Amazon S3 en Outposts

Los objetos son las entidades fundamentales almacenadas en Amazon S3 en Outposts. Cada objeto está almacenado en un bucket. Debe utilizar puntos de acceso para acceder a cualquier objeto de un bucket de Outpost. Cuando especifica el bucket para las operaciones de objetos, se utiliza el Nombre de recurso de Amazon (ARN) del punto de acceso o el alias del punto de acceso. Para obtener más información acerca de los alias de punto de acceso, consulte Uso de un alias de estilo de bucket para su punto de acceso de bucket de S3 en Outposts.

En el siguiente ejemplo se muestra el formato ARN para los puntos de acceso de S3 en Outposts, que incluye el código Región de AWS de la Región a la que pertenece el Outpost, el ID de Cuenta de AWS, el ID de Outposts y el nombre del punto de acceso:

```
arn:aws:s3-outposts:region:account-id:outpost/outpost-id/accesspoint/accesspoint-name
```

Para obtener más información acerca de S3 en Outposts, consulte ARN de recursos para S3 en Outposts.

Con Amazon S3 en Outposts, los datos de objeto siempre se almacenan en el Outpost. Cuando AWS instala un bastidor de Outpost, sus datos permanecen de manera local en su Outpost para cumplir los requisitos de residencia de datos. Sus objetos nunca salen de su Outpost y no están en una Región de AWS. Ya que la AWS Management Console está alojada dentro de la región, no puede usar la consola para cargar o administrar objetos en su Outpost. Sin embargo, puede utilizar la API de REST, AWS Command Line Interface (AWS CLI) y los SDK de AWS para cargar y administrar los objetos a través de los puntos de acceso.

En los siguientes ejemplos, se muestra cómo eliminar un solo objeto o varios objetos en un bucket de S3 en Outposts con AWS Command Line Interface (AWS CLI) y AWS SDK for Java.

Mediante AWS CLI

En los ejemplos siguientes, se muestra cómo eliminar un solo objeto o varios objetos de un bucket de S3 en Outposts.

delete-object

En el siguiente ejemplo, se elimina un objeto denominado `sample-object.xml` de un bucket de S3 en Outposts (`s3-outposts:DeleteObject`) mediante la AWS CLI. Para usar este comando, sustituya *`user input placeholder`* por su propia información. Para obtener más información sobre este comando, consulte delete-object en la Referencia de AWS CLI.

```
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

En el siguiente ejemplo, se eliminan dos objetos denominados `sample-object.xml` y `test1.text` de un bucket de S3 en Outposts (`s3-outposts:DeleteObject`) mediante la AWS CLI. Para usar este comando, sustituya *`user input placeholder`* por su propia información. Para obtener más información sobre este comando, consulte delete-objects en la Referencia de AWS CLI.

```
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
}
```

Uso de AWS SDK para Java

En los ejemplos siguientes, se muestra cómo eliminar un solo objeto o varios objetos de un bucket de S3 en Outposts.

DeleteObject

En el siguiente ejemplo de S3 en Outposts, se elimina un objeto de un bucket mediante el SDK para Java. Para utilizar este ejemplo, especifique el ARN del punto de acceso para Outpost y el nombre de la clave del objeto que desea eliminar. Para obtener más información, consulte DeleteObject en la Referencia de la API de 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

En el siguiente ejemplo de S3 en Outposts, se cargan y luego, eliminan objetos de un bucket mediante el SDK para Java. Para utilizar este ejemplo, especifique el ARN del punto de acceso para Outpost. Para obtener más información, consulte DeleteObjects en la Referencia de la API de 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.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();
        }
    }
}
```

Uso de HeadBucket para determinar si existe un bucket de S3 en Outposts y si tiene permisos de acceso

Los objetos son las entidades fundamentales almacenadas en Amazon S3 en Outposts. Cada objeto está almacenado en un bucket. Debe utilizar puntos de acceso para acceder a cualquier objeto de un bucket de Outpost. Cuando especifica el bucket para las operaciones de objetos, se utiliza el Nombre de recurso de Amazon (ARN) del punto de acceso o el alias del punto de acceso. Para obtener más información acerca de los alias de punto de acceso, consulte Uso de un alias de estilo de bucket para su punto de acceso de bucket de S3 en Outposts.

En el siguiente ejemplo se muestra el formato ARN para los puntos de acceso de S3 en Outposts, que incluye el código Región de AWS de la Región a la que pertenece el Outpost, el ID de Cuenta de AWS, el ID de Outposts y el nombre del punto de acceso:

```
arn:aws:s3-outposts:region:account-id:outpost/outpost-id/accesspoint/accesspoint-name
```

Para obtener más información acerca de S3 en Outposts, consulte ARN de recursos para S3 en Outposts.

> **Note**
>
> Con Amazon S3 en Outposts, los datos de objeto siempre se almacenan en el Outpost. Cuando AWS instala un bastidor de Outpost, sus datos permanecen de manera local en su Outpost para cumplir los requisitos de residencia de datos. Sus objetos nunca salen de su Outpost y no están en una Región de AWS. Ya que la AWS Management Console está alojada dentro de la región, no puede usar la consola para cargar o administrar objetos en su Outpost. Sin embargo, puede utilizar la API de REST, AWS Command Line Interface (AWS CLI) y los SDK de AWS para cargar y administrar los objetos a través de los puntos de acceso.

En los siguientes ejemplos de AWS Command Line Interface (AWS CLI) y AWS SDK for Java, se muestra cómo usar la operación de la API HeadBucket para determinar si existe un bucket de S3 en Outposts y si tiene permiso para acceder a él. Para obtener más información, consulte HeadBucket en la Referencia de la API de Amazon Simple Storage Service.

Mediante AWS CLI

En el siguiente ejemplo de S3 en Outposts de AWS CLI, se utiliza el comando `head-bucket` para determinar si existe un bucket y si tiene permiso para acceder a él. Para usar este comando, sustituya *`user input placeholder`* por su propia información. Para obtener más información acerca de este comando, consulte head-bucket en la Referencia de AWS CLI.

```
aws s3api head-bucket --bucket arn:aws:s3-
outposts:region:123456789012:outpost/op-01ac5d28a6a232904/accesspoint/example-outposts-
access-point
```

Uso de AWS SDK para Java

En el siguiente ejemplo de S3 en Outposts, se muestra cómo determinar si existe un bucket y si usted tiene permiso para acceder a él. Para utilizar este ejemplo, especifique el ARN del punto de acceso para Outpost. Para obtener más información, consulte HeadBucket en la Referencia de la API de 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.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();
        }
    }
}
```

Ejecución y administración de una carga multiparte con el SDK para Java

Con Amazon S3 en Outposts, puede crear buckets de S3 en recursos de AWS Outposts y almacenar y recuperar objetos en las instalaciones para las aplicaciones que requieren acceso local a los datos, procesamiento local de los datos y residencia de los datos. Puede utilizar S3 en Outposts a través de la AWS Management Console, AWS Command Line Interface (AWS CLI), AWS SDK o la API de REST. Para obtener más información, consulte ¿Qué es Amazon S3 en Outposts?

En los siguientes ejemplos, se muestra cómo puede utilizar S3 en Outposts con AWS SDK for Java para realizar y administrar una carga multiparte.

Temas

- Realización de una carga multiparte de un objeto en un bucket de S3 en Outposts
- Copia de un objeto grande en un bucket de S3 en Outposts con la carga multiparte
- Obtención de una lista de las partes de un objeto en un bucket de S3 en Outposts
- Recuperación de una lista de cargas multiparte en curso en un bucket de S3 en Outposts

Realización de una carga multiparte de un objeto en un bucket de S3 en Outposts

En el siguiente ejemplo de S3 en Outposts, se inicia, carga y finaliza una carga multiparte de un objeto en un bucket mediante el SDK para Java. Para utilizar este ejemplo, sustituya *`user input placeholder`* por su propia información. Para obtener más información, consulte Carga de un objeto con la carga multiparte .

```
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;
    }
```

Copia de un objeto grande en un bucket de S3 en Outposts con la carga multiparte

En el siguiente ejemplo de S3 en Outposts se utiliza el SDK para Java para copiar un objeto en un bucket. Para utilizar este ejemplo, sustituya *user input placeholder* por su propia información. Este ejemplo está adaptado de Copiar un objeto con la carga multiparte.

```
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;
    }
}
```

Obtención de una lista de las partes de un objeto en un bucket de S3 en Outposts

En el siguiente ejemplo de S3 en Outposts se muestran las partes de un objeto en un bucket con el SDK para Java. Para utilizar este ejemplo, sustituya *`user input placeholder`* por su propia información.

```
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();
        }
    }
}
```

Recuperación de una lista de cargas multiparte en curso en un bucket de S3 en Outposts

En el siguiente ejemplo de Amazon S3 en Outposts, se muestra cómo recuperar una lista de cargas multiparte en curso desde un bucket de Outposts mediante el SDK para Java. Para utilizar este ejemplo, sustituya *`user input placeholder`* por su propia información. Este es un ejemplo adaptado del ejemplo Descripción de cargas multiparte para 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();
        }
    }
}
```

Uso de URL prefirmadas para S3 en Outposts

Para conceder acceso por tiempo limitado a los objetos que se almacenan localmente en un Outpost sin actualizar su política de bucket, puede usar una URL prefirmada. Con las URL prefirmadas, usted, como propietario del bucket, puede compartir objetos con personas en su nube privada virtual (VPC) o concederles la capacidad de cargar o eliminar objetos.

Cuando crea una URL prefirmada con el SDK de AWS o el AWS Command Line Interface (AWS CLI), asocia la URL a una acción específica. También puede conceder acceso por tiempo limitado a la URL prefirmada eligiendo un tiempo de caducidad personalizado que puede ser de tan solo

1 segundo y de hasta 7 días. Cuando comparte la URL prefirmada, la persona de la VPC puede realizar la acción incrustada en la URL como si fuera el usuario de firma original. La URL caducará y ya no funcionará cuando llegue a su hora de vencimiento.

Limitación de las capacidades de URL prefirmadas

Las capacidades de una URL están limitadas por los permisos del usuario que la creó. En esencia, las URL prefirmadas son tokens al portador que otorgan acceso a quienes las poseen. Por lo tanto, le recomendamos que los proteja adecuadamente.

AWS Signature Version 4 (SigV4)

Para aplicar un comportamiento específico cuando las solicitudes de URL prefirmadas se autentican mediante AWS Signature Version 4 (SigV4), puede usar claves de condición en las políticas de bucket y en las políticas de punto de acceso. Por ejemplo, puede crear una política de bucket que use la condición `s3-outposts:signatureAge` para denegar cualquier solicitud de URL prefirmada de Amazon S3 en Outposts en los objetos del bucket `example-outpost-bucket` si la firma tiene más de 10 minutos de antigüedad. Para utilizar este ejemplo, reemplace los *`user input placeholders`* con su propia información.

```
{
    "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"}
            }
        }
    ]
}
```

Para obtener una lista de claves de condición y políticas de ejemplo adicionales que puede usar para imponer un comportamiento específico cuando las solicitudes de URL prefirmadas se autentican mediante Signature Version 4, consulte Claves de política de autenticación de AWS Signature Version 4 (SigV4).

Restricción de ruta de red

Si desea restringir el uso de direcciones URL prefirmadas y todo el acceso de S3 en Outposts a rutas de red concretas, puede escribir políticas que requieran una ruta de red determinada. Para establecer la restricción en la entidad principal de IAM que realiza la llamada, puede usar políticas de AWS Identity and Access Management (IAM) basadas en identidades (por ejemplo, políticas de usuario, grupo o rol). Para establecer la restricción en el recurso S3 en Outposts, puede usar políticas basadas en recursos (por ejemplo, políticas de bucket y punto de acceso).

Una restricción de ruta de red en la entidad principal de IAM requiere que el usuario de esas credenciales realice solicitudes desde la red especificada. Una restricción en el bucket o en el punto de acceso requiere que todas las solicitudes a ese recurso se originen desde la red especificada. Estas restricciones también se aplican fuera del escenario de URL prefirmada.

La condición global de IAM que utilice depende del tipo de punto de conexión. Si está utilizando el punto de conexión público para S3 en Outposts, utilice `aws:SourceIp`. Si utiliza un punto de conexión de VPC en S3 en Outposts, utilice `aws:SourceVpc` o `aws:SourceVpce`.

La siguiente instrucción de política de IAM requiere que la entidad principal acceda a AWS solo desde el rango de red especificado. Con esta declaración de política, todo acceso debe originarse desde ese rango. Esto incluye el caso de alguien que usa una URL prefirmada para S3 en Outposts. Para utilizar este ejemplo, reemplace los *`user input placeholders`* con su propia información.

```
{
    "Sid": "NetworkRestrictionForIAMPrincipal",
    "Effect": "Deny",
    "Action": "*",
    "Resource": "*",
    "Condition": {
        "NotIpAddressIfExists": {"aws:SourceIp": "IP-address-range"},
        "BoolIfExists": {"aws:ViaAWSService": "false"}
    }
}
```

Para ver una política de bucket de ejemplo que usa la clave de condición global `aws:SourceIP` de AWS para restringir el acceso a un bucket de S3 en Outposts a un rango de red específico, consulte Configuración de IAM con S3 en Outposts.

Quién puede crear una URL prefirmada

Cualquiera que tenga credenciales de seguridad válidas puede crear una URL prefirmada. Sin embargo, para que un usuario de la VPC pueda acceder a un objeto correctamente, la URL prefirmada debe haber sido creada por alguien que tenga permiso para realizar la operación en la que se basa la URL prefirmada.

Puede usar estas credenciales para crear una URL prefirmada:

- Perfil de instancia de IAM: válido hasta 6 horas.
- AWS Security Token Service: válido hasta 36 horas cuando se firma con las credenciales permanentes, como, por ejemplo, las credenciales del usuario raíz de la Cuenta de AWS o un usuario de IAM.
- Usuario de IAM: válido hasta 7 días cuando se utiliza AWS Signature Version 4.

  Para crear una URL prefirmada que es válida hasta 7 días, primero delegue las credenciales de usuario de IAM (la clave de acceso y la clave secreta) al SDK que está utilizando. A continuación, genere una URL prefirmada mediante AWS Signature Version 4.

> **Note**
>
> - Si creó una URL prefirmada con un token temporal, la URL caducará cuando caduque el token, incluso si creó la URL con un tiempo de vencimiento posterior.
> - Dado que las URL prefirmadas otorgan acceso a los buckets de S3 en Outposts a quien tenga la URL, recomendamos que los proteja adecuadamente. Para obtener más información sobre la protección de las URL prefirmadas, consulte Limitación de las capacidades de URL prefirmadas.

¿Cuándo comprueba S3 en Outposts la fecha y hora de vencimiento de una URL prefirmada?

S3 en Outposts comprueba la fecha y hora de vencimiento de una URL firmada al realizarse la solicitud HTTP. Por ejemplo, si un cliente comienza a descargar un archivo grande inmediatamente antes de la fecha de vencimiento, la descarga continúa incluso si se sobrepasa la hora de vencimiento durante la descarga. Sin embargo, si la conexión se interrumpe y el cliente intenta reiniciar la descarga después de la hora de vencimiento, la descarga produce un error.

Para obtener más información sobre el uso de una URL prefirmada con objeto de compartir o cargar objetos, consulte los siguientes temas.

Temas

- Uso compartido de objetos con URL prefirmadas
- Generación de una URL prefirmada para cargar un objeto en un bucket de S3 en Outposts

Uso compartido de objetos con URL prefirmadas

Para conceder acceso por tiempo limitado a los objetos que se almacenan localmente en un Outpost sin actualizar su política de bucket, puede usar una URL prefirmada. Con las URL prefirmadas, usted, como propietario del bucket, puede compartir objetos con personas en su nube privada virtual (VPC) o concederles la capacidad de cargar o eliminar objetos.

Cuando crea una URL prefirmada con el SDK de AWS o el AWS Command Line Interface (AWS CLI), asocia la URL a una acción específica. También puede conceder acceso por tiempo limitado a la URL prefirmada eligiendo un tiempo de caducidad personalizado que puede ser de tan solo 1 segundo y de hasta 7 días. Cuando comparte la URL prefirmada, la persona de la VPC puede realizar la acción incrustada en la URL como si fuera el usuario de firma original. La URL caducará y ya no funcionará cuando llegue a su hora de vencimiento.

Cuando crea una URL prefirmada, debe proporcionar sus credenciales de seguridad y luego especificar lo siguiente:

- Un nombre de recurso de Amazon (ARN) de punto de acceso para el bucket de Amazon S3 en Outposts
- Una clave del objeto

- Un método HTTP (`GET` para descargar objetos)
- Una fecha y hora de caducidad

Una URL prefirmada solo es válida para la duración especificada. Es decir, debe comenzar la acción permitida por la URL antes de la fecha y hora de vencimiento. Puede utilizar una URL prefirmada varias veces, hasta la fecha y hora de vencimiento. Si creó una URL prefirmada con un token temporal, la URL caducará cuando caduque el token, incluso si creó la URL con un tiempo de vencimiento posterior.

Los usuarios de la nube privada virtual (VPC) que tienen acceso a la URL prefirmada pueden acceder al objeto. Por ejemplo, si tiene un video en su bucket y tanto el bucket como el objeto son privados, puede compartir el video con otros generando una URL prefirmada. Dado que las URL prefirmadas otorgan acceso a sus buckets de S3 en Outposts a quien tenga la URL, recomendamos que las proteja adecuadamente. Para obtener más información acerca de la protección de direcciones URL prefirmadas, consulte Limitación de las capacidades de URL prefirmadas.

Cualquiera que tenga credenciales de seguridad válidas puede crear una URL prefirmada. Sin embargo, la URL prefirmada debe haber sido creada por alguien que tenga permisos para realizar la operación en la que se basa la URL prefirmada. Para obtener más información, consulte Quién puede crear una URL prefirmada.

Puede generar una URL prefirmada para compartir un objeto en un bucket de S3 en Outposts mediante el SDK de AWS y la AWS CLI. Para obtener más información, consulte los ejemplos siguientes.

Uso de los AWS SDK

Puede usar los SDK de AWS para generar una URL prefirmada que puede dar a terceros para que puedan recuperar un objeto.

> **Note**
>
> Cuando use los SDK de AWS para generar una URL prefirmada, el tiempo máximo de vencimiento de una URL prefirmada es de 7 días desde el momento de su creación.

Java

Example

El siguiente ejemplo genera una URL prefirmada que puede dar a terceros de modo que puedan recuperar un objeto desde un bucket de S3 en Outposts. Para obtener más información, consulte Uso de URL prefirmadas para S3 en Outposts. Para utilizar este ejemplo, reemplace los *user input placeholders* con su propia información.

Para obtener instrucciones sobre cómo crear y probar una muestra funcional, consulte Introducción en la Guía del desarrollador de AWS SDK for Java.

```
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

El siguiente ejemplo genera una URL prefirmada que puede dar a terceros de modo que puedan recuperar un objeto desde un bucket de S3 en Outposts. Para obtener más información, consulte Uso de URL prefirmadas para S3 en Outposts. Para utilizar este ejemplo, reemplace los *`user input placeholders`* con su propia información.

Para obtener información acerca de cómo configurar y ejecutar ejemplos de código, consulte Introducción al SDK de AWS para .NET en la Guía para desarrolladores del SDK de AWS para .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

Los siguientes ejemplos generan una URL prefirmada para compartir un objeto mediante el SDK para Python (Boto3). Por ejemplo, utilice un cliente Boto3 y la función `generate_presigned_url` para generar una URL prefirmada que le permita `GET` un objeto.

```
import boto3
    url = boto3.client('s3').generate_presigned_url(
    ClientMethod='get_object',
    Params={'Bucket': 'ACCESS_POINT_ARN', 'Key': 'OBJECT_KEY'},
    ExpiresIn=3600)
```

Con el objetivo de obtener más información acerca del uso de SDK para Python (Boto3) a fin de generar una URL prefirmada, consulte Python en la Referencia de la API de AWS SDK for Python (Boto).

Uso de la AWS CLI

El siguiente ejemplo del comando de AWS CLI genera una URL prefirmada para un bucket de S3 en Outposts. Para utilizar este ejemplo, reemplace los *`user input placeholders`* con su propia información.

> **Note**
>
> Cuando use la AWS CLI para generar una URL prefirmada, el tiempo máximo de vencimiento de una URL prefirmada es de 7 días desde el momento de su creación.

```
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
```

Para obtener más información, consulte presign en la Referencia de comandos de la AWS CLI.

Generación de una URL prefirmada para cargar un objeto en un bucket de S3 en Outposts

Para conceder acceso por tiempo limitado a los objetos que se almacenan localmente en un Outpost sin actualizar su política de bucket, puede usar una URL prefirmada. Con las URL prefirmadas,

usted, como propietario del bucket, puede compartir objetos con personas en su nube privada virtual (VPC) o concederles la capacidad de cargar o eliminar objetos.

Cuando crea una URL prefirmada con el SDK de AWS o el AWS Command Line Interface (AWS CLI), asocia la URL a una acción específica. También puede conceder acceso por tiempo limitado a la URL prefirmada eligiendo un tiempo de caducidad personalizado que puede ser de tan solo 1 segundo y de hasta 7 días. Cuando comparte la URL prefirmada, la persona de la VPC puede realizar la acción incrustada en la URL como si fuera el usuario de firma original. La URL caducará y ya no funcionará cuando llegue a su hora de vencimiento.

Cuando crea una URL prefirmada, debe proporcionar sus credenciales de seguridad y luego especificar lo siguiente:

- Un nombre de recurso de Amazon (ARN) de punto de acceso para el bucket de Amazon S3 en Outposts
- Una clave del objeto
- Un método HTTP (PUT para cargar objetos)
- Una fecha y hora de caducidad

Una URL prefirmada solo es válida para la duración especificada. Es decir, debe comenzar la acción permitida por la URL antes de la fecha y hora de vencimiento. Puede utilizar una URL prefirmada varias veces, hasta la fecha y hora de vencimiento. Si creó una URL prefirmada con un token temporal, la URL caducará cuando caduque el token, incluso si creó la URL con un tiempo de vencimiento posterior.

Si la acción permitida por una URL prefirmada consta de varios pasos, como una carga multiparte, todos los pasos deben comenzar antes de la hora de vencimiento. Si S3 en Outposts intenta comenzar un paso con una URL vencida, recibirá un error.

Los usuarios de la nube privada virtual (VPC) que tienen acceso a la URL prefirmada pueden cargar objetos. Por ejemplo, un usuario de la VPC que tenga acceso a la URL prefirmada puede cargar un objeto en su bucket. Dado que las URL prefirmadas otorgan acceso a su bucket de S3 en Outposts a cualquier usuario de la VPC que tenga acceso a la URL prefirmada, recomendamos que las proteja adecuadamente. Para obtener más información acerca de la protección de direcciones URL prefirmadas, consulte Limitación de las capacidades de URL prefirmadas.

Cualquiera que tenga credenciales de seguridad válidas puede crear una URL prefirmada. Sin embargo, la URL prefirmada debe haber sido creada por alguien que tenga permisos para realizar

la operación en la que se basa la URL prefirmada. Para obtener más información, consulte Quién puede crear una URL prefirmada .

Genere una URL prefirmada para una operación de objeto de S3 en Outposts mediante los SDK de AWS

Java

SDK para Java 2.x

En este ejemplo, se muestra cómo generar una URL prefirmada que puede usar para cargar un objeto en un bucket de S3 en Outposts durante un tiempo limitado. Para obtener más información, consulte Uso de URL prefirmadas para S3 en 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 para Python (Boto3)

En este ejemplo, se muestra cómo generar una URL prefirmada que pueda realizar una acción de S3 en Outposts durante un tiempo limitado. Para obtener más información, consulte Uso de URL prefirmadas para S3 en Outposts . Para realizar una solicitud con la URL, utilice el paquete `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 en Outposts con Amazon EMR en Outposts local

Amazon EMR es una plataforma de clúster administrada que simplifica la ejecución de marcos de macrodatos, tales como Apache Hadoop y Apache Spark en AWS para procesar y analizar grandes cantidades de datos. Mediante el uso de estos marcos de trabajo y proyectos de código abierto relacionados, puede procesar datos para fines de análisis y cargas de trabajo de inteligencia empresarial. Además, Amazon EMR permite transformar y trasladar grandes cantidades de datos hacia y desde otros almacenes y bases de datos de AWS, como Amazon S3 en Outposts. Para

obtener más información sobre Amazon EMR, consulte Clústeres de EMR en AWS Outposts en la Guía de administración de Amazon EMR.

Para Amazon S3 en Outposts, Amazon EMR comenzó a admitir el conector S3A de Apache Hadoop en la versión 7.0.0. Las versiones anteriores de Amazon EMR no admiten S3 en Outposts localmente y tampoco son compatibles con el sistema de archivos de EMR (EMRFS).

Aplicaciones compatibles

Amazon EMR con Amazon S3 en Outposts admite las siguientes aplicaciones:

- Hadoop
- Spark
- Hue
- Hive
- Sqoop
- Pig
- Hudi
- Flink

Para obtener más información, consulte la Guía de publicación de Amazon EMR.

Creación y configuración de un bucket de Amazon S3 en Outposts

Amazon EMR utiliza el AWS SDK for Java con Amazon S3 en Outposts para almacenar datos de entrada y de salida. Los archivos de registro de Amazon EMR se almacenan en la ubicación regional de Amazon S3 que elija pero no se almacenan localmente en Outpost. Para obtener más información, consulte Ver archivos de registro en la Guía de administración de Amazon EMR.

Los buckets de S3 on Outposts aplican ciertas restricciones y limitaciones de nomenclatura para cumplir con los requisitos de Amazon S3 y DNS. Para obtener más información, consulte Creación de un bucket de S3 en Outposts.

Con la versión 7.0.0 y posteriores de Amazon EMR, puede usar Amazon EMR con S3 en Outposts y el sistema de archivos S3A.

Requisitos previos

Permisos de S3 en Outposts: al crear el perfil de instancia de Amazon EMR, su rol debe incluir el espacio de nombres de AWS Identity and Access Management (IAM) para S3 en Outposts. S3 en Outposts tiene su propio espacio de nombres: `s3-outposts*`. Para ver un ejemplo de política que utiliza este espacio de nombres, consulte Configuración de IAM con S3 en Outposts.

Conector S3A: para configurar el clúster de EMR para que pueda acceder a los datos de un bucket de Amazon S3 en Outposts, debe utilizar el conector S3A de Apache Hadoop. Para usar el conector, asegúrese de que todos sus URI de S3 usen el esquema `s3a`. Si no es así, puede configurar la implementación del sistema de archivos que utiliza para el clúster de EMR para que sus URI de S3 funcionen con el conector S3A.

Para configurar la implementación del sistema de archivos para que funcione con el conector S3A, utilice las propiedades de configuración `fs.`*`file_scheme`*`.impl` y `fs.AbstractFileSystem.`*`file_scheme`*`.impl` del clúster de EMR, donde *`file_scheme`* equivale al tipo de URI de S3 que tenga. Para utilizar el ejemplo siguiente, sustituya *`user input placeholders`* con su propia información. Por ejemplo, para cambiar la implementación del sistema de archivos para los URI de S3 que utilizan el esquema `s3`, especifique las siguientes propiedades de configuración del clúster:

```
[
  {
"Classification": "core-site",
    "Properties": {
    "fs.s3.impl": "org.apache.hadoop.fs.s3a.S3AFileSystem",
    "fs.AbstractFileSystem.s3.impl": "org.apache.hadoop.fs.s3a.S3A"
    }
  }
]
```

Para usar S3A, defina la propiedad de configuración `fs.`*`file_scheme`*`.impl` en `org.apache.hadoop.fs.s3a.S3AFileSystem` y establezca la propiedad `fs.AbstractFileSystem.`*`file_scheme`*`.impl` en `org.apache.hadoop.fs.s3a.S3A`.

Por ejemplo, si accede a la ruta `s3a://`*`bucket`*`/...`, defina la propiedad `fs.s3a.impl` en `org.apache.hadoop.fs.s3a.S3AFileSystem` y establezca la propiedad `fs.AbstractFileSystem.s3a.impl` en `org.apache.hadoop.fs.s3a.S3A`.

Introducción al uso de Amazon EMR con Amazon S3 en Outposts

En los temas que siguen se explica cómo empezar a utilizar Amazon EMR con Amazon S3 en Outposts.

Temas

Crear una política de permisos

Antes de poder crear un clúster de EMR que utilice Amazon S3 en Outposts, debe crear una política de IAM para adjuntarla al perfil de instancia de Amazon EC2 para el clúster. La política debe tener permisos de acceso al Nombre de recurso de Amazon (ARN) del punto de acceso de S3 en Outposts. Para obtener más información acerca de la creación de políticas de IAM para S3 en Outposts, consulte Configuración de IAM con S3 en Outposts.

En la siguiente política de ejemplo se muestra cómo conceder los permisos necesarios. Después de crear la política, adjúntela al rol de perfil de instancia que utilice para crear su clúster de EMR, tal y como se describe en la sección the section called “Creación y configuración de un clúster”. Para utilizar este ejemplo, reemplace los *user input placeholders* con su propia información.

```
{
"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:*"
            ]
        }
    ]

 }
```

Creación y configuración de un clúster

Para crear un clúster que ejecute Spark con S3 en Outposts, complete los siguientes pasos en la consola.

Para crear un clúster que ejecute Spark con S3 en Outposts

1. Abra la consola de Amazon EMR enhttps://console.aws.amazon.com/elasticmapreduce/.
2. En el panel de navegación izquierdo, elija Clusters (Clústeres).
3. Elija Create cluster.
4. Para la versión de Amazon EMR, elija emr-7.0.0 o posterior.
5. Para el paquete de aplicaciones, elija Interactivo con Spark. Seleccione cualquier otra aplicación que desee incluir en el clúster.
6. Para habilitar Amazon S3 en Outposts, realice la siguiente configuración.

   Ejemplo de configuración

   Para usar esta configuración de ejemplo, sustituya *user input placeholders* por su información.

```
[
 {
   "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. En la sección Redes, elija una nube privada virtual (VPC) y una subred que estén en su bastidor de AWS Outposts. Para obtener más información sobre Amazon EMR en Outposts, consulte Clústeres de EMR en AWS Outposts en la Guía de administración de Amazon EMR.
8. En la sección Perfil de instancia de EC2 para Amazon EMR, elija el rol de IAM que tenga adjunta la política de permisos que ha creado anteriormente.
9. Configure los ajustes de clúster restantes y, a continuación, elija Crear clúster.

Información general sobre las configuraciones

En las siguientes tablas se describen las configuraciones de S3A y Spark, además de los valores que se deben especificar para los parámetros al configurar un clúster que utiliza S3 en Outposts con Amazon EMR.

Configuración de S3A

Parámetro	Valor predeterminado	Valor obligatorio para S3 en Outposts	Explicación
`fs.s3a.aws.credentials.provider`	Si no se especifica, S3A buscará el bucket de S3 de la región con el nombre del bucket de Outposts.	El ARN del punto de acceso del bucket de S3 en Outposts	Amazon S3 en Outposts admite puntos de acceso únicamente de la virtual private cloud (VPC) como el único medio para acceder a los buckets de Outposts.
`fs.s3a.committer.name`	`file`	`magic`	Magic es el único confirmador compatible con S3 en Outposts.
`fs.s3a.select.enabled`	`TRUE`	`FALSE`	S3 Select no es compatible con Outposts.
`JAVA_HOME`	`/usr/lib/jvm/java-8`	`/usr/lib/jvm/java-11-amazon-corretto.x86_64`	S3 en Outposts en S3A requiere la versión 11 de Java.

Configuración de Spark

Parámetro	Valor predeterminado	Valor obligatorio para S3 en Outposts	Explicación
`spark.sql.sources.fastS3PartitionDiscovery.enabled`	`TRUE`	`FALSE`	S3 en Outposts no admite la partición rápida.

Parámetro	Valor predeterminado	Valor obligatorio para S3 en Outposts	Explicación
`spark.executorEnv.JAVA_HOME`	`/usr/lib/jvm/java-8`	`/usr/lib/jvm/java-11-amazon-corretto.x86_64`	S3 en Outposts en S3A requiere la versión 11 de Java.

Consideraciones

Tenga en cuenta lo siguiente cuando integre Amazon EMR con los buckets de S3 en Outposts:

- Amazon S3 en Outposts es compatible con la versión 7.0.0 y posteriores de Amazon EMR.
- Se requiere el conector S3A para utilizar S3 en Outposts con Amazon EMR. Solo S3A tiene las características necesarias para interactuar con los buckets de S3 en Outposts. Para obtener información sobre la configuración del conector S3A, consulte el apartado Requisitos previos.
- Amazon S3 en Outposts solo admite el cifrado del servidor con claves administradas por Amazon S3 (SSE-S3) con Amazon EMR. Para obtener más información, consulte the section called “Cifrado de datos”.
- Amazon S3 en Outposts no admite la escritura con el FileOutputCommitter de S3A. Al escribir con el FileOutputCommitter de S3A en los buckets de S3 en Outposts, se produce el siguiente error: InvalidStorageClass: The storage class you specified is not valid.
- Amazon S3 en Outposts no es compatible con Amazon EMR sin servidor ni Amazon EMR en EKS.
- Los registros de Amazon EMR se almacenan en la ubicación regional de Amazon S3 que haya elegido pero no se almacenan localmente en el bucket de S3 en Outposts.

Almacenamiento en caché de autorización y autenticación

S3 en Outposts almacena en caché localmente los datos de autenticación y autorización de forma segura en los bastidores de Outposts. La memoria caché elimina viajes de ida y vuelta a la Región de AWS principal por cada solicitud. Esto elimina la variabilidad que generan los viajes de ida y vuelta en la red. Con la caché de autenticación y autorización de S3 en Outposts, obtiene latencias consistentes que son independientes de la latencia de la conexión entre Outposts y la Región de AWS.

Cuando realiza una solicitud a la API de S3 en Outposts, los datos de autenticación y autorización se almacenan en caché de forma segura. Luego, los datos en caché se utilizan para autenticar las solicitudes posteriores a la API de objetos de S3. S3 en Outposts solo almacena en caché los datos de autenticación y autorización cuando la solicitud se firma utilizando Signature Version 4A (SigV4A). La caché se almacena localmente en los Outposts dentro del servicio S3 en Outposts. Se actualiza de forma asíncrona cuando se realiza una solicitud a la API de S3. La caché se cifra y en Outposts no se almacena ninguna clave criptográfica en texto sin formato.

La caché es válida durante un máximo de 10 minutos cuando el Outpost está conectado a la Región de AWS. Se actualiza de forma asíncrona cuando realiza una solicitud a la API de S3 en Outposts para garantizar que se utilicen las políticas más recientes. Si el Outpost se desconecta de la Región de AWS, la caché será válida durante un máximo de 12 horas.

Configuración de la caché de autorización y autenticación

S3 en Outposts almacena automáticamente en caché los datos de autenticación y autorización de las solicitudes firmadas con el algoritmo SigV4A. Para obtener más información, consulte Firma de solicitudes de API de AWS en la Guía del usuario de AWS Identity and Access Management. El algoritmo SigV4A está disponible en las versiones más recientes de los SDK de AWS. Puede obtenerlo a través de una dependencia en las bibliotecas de AWS Common Runtime (CRT).

Debe usar la versión más reciente del SDK de AWS e instalar la versión más reciente de CRT. Por ejemplo, puede ejecutar `pip install awscrt` para obtener la versión más reciente de CRT con Boto3.

S3 en Outposts no almacena en caché los datos de autenticación y autorización de las solicitudes firmadas con el algoritmo SigV4.

Validación de la firma de SigV4

Se puede utilizar AWS CloudTrail para validar que las solicitudes se hayan firmado con SigV4. Para obtener más información sobre la configuración de CloudTrail para S3 en Outposts, consulte Monitoreo de S3 en Outposts con registros de AWS CloudTrail.

Tras configurar CloudTrail, puede comprobar cómo se firmó una solicitud en el campo `SignatureVersion` de los registros de CloudTrail. Las solicitudes que se hayan firmado con SigV4A tendrán `SignatureVersion` establecido en `AWS4-ECDSA-P256-SHA256`. Las solicitudes que se hayan firmado con SigV4 tendrán `SignatureVersion` establecido en `AWS4-HMAC-SHA256`.

Seguridad en S3 en Outposts

La seguridad en la nube de AWS es la mayor prioridad. Como cliente de AWS, se beneficiará de una arquitectura de red y de centros de datos diseñados para satisfacer los requisitos de seguridad de las organizaciones más exigentes.

La seguridad es una responsabilidad compartida entre AWS y usted. El modelo de responsabilidad compartida la describe como seguridad de la nube y seguridad en la nube:

- Seguridad de la nube: AWS es responsable de proteger la infraestructura que ejecuta Servicios de AWS en Nube de AWS. Además, AWS proporciona servicios que puede utilizar de forma segura. Los auditores externos prueban y verifican periódicamente la eficacia de nuestra seguridad como parte de los AWSProgramas de conformidad de . Para obtener información sobre los programas de conformidad que se aplican a Amazon S3 en Outposts, consulte Servicios de AWS en el ámbito del programa de conformidad.
- Seguridad en la nube: su responsabilidad viene determinada por el Servicio de AWS que utilice. También es responsable de otros factores, incluida la confidencialidad de los datos, los requisitos de la empresa y la legislación y los reglamentos vigentes.

Esta documentación le ayuda a comprender cómo aplicar el modelo de responsabilidad compartida cuando se utiliza S3 en Outposts. En los siguientes temas, se le mostrará cómo configurar S3 en Outposts para satisfacer sus objetivos de seguridad y conformidad. También puede obtener información sobre cómo utilizar otros Servicios de AWS que le ayuden a monitorear y proteger los recursos de S3 en Outposts.

Temas

Cifrado de datos en S3 en Outposts

De forma predeterminada, todos los datos almacenados en Amazon S3 en Outposts se cifran mediante cifrado del lado del servidor con claves de cifrado administradas de Amazon S3 (SSE-S3).

Para obtener más información, consulte Uso del cifrado del servidor con claves administradas por Amazon S3 (SSE-S3) .

Opcionalmente, puede usar el cifrado del lado del servidor con claves proporcionadas por el cliente (SSE-C). Para utilizar SSE-C, especifique una clave de cifrado como parte de las solicitudes de API de objeto. El cifrado en el servidor solo cifra los datos de objetos, no los metadatos de objetos. Para obtener más información, consulte Uso de cifrado en el lado del servidor con claves proporcionadas por el cliente (SSE-C) .

> **Note**
>
> S3 en Outposts no es compatible con el cifrado del lado del servidor con claves AWS Key Management Service (AWS KMS) (SSE-KMS).

AWS PrivateLink para S3 en Outposts

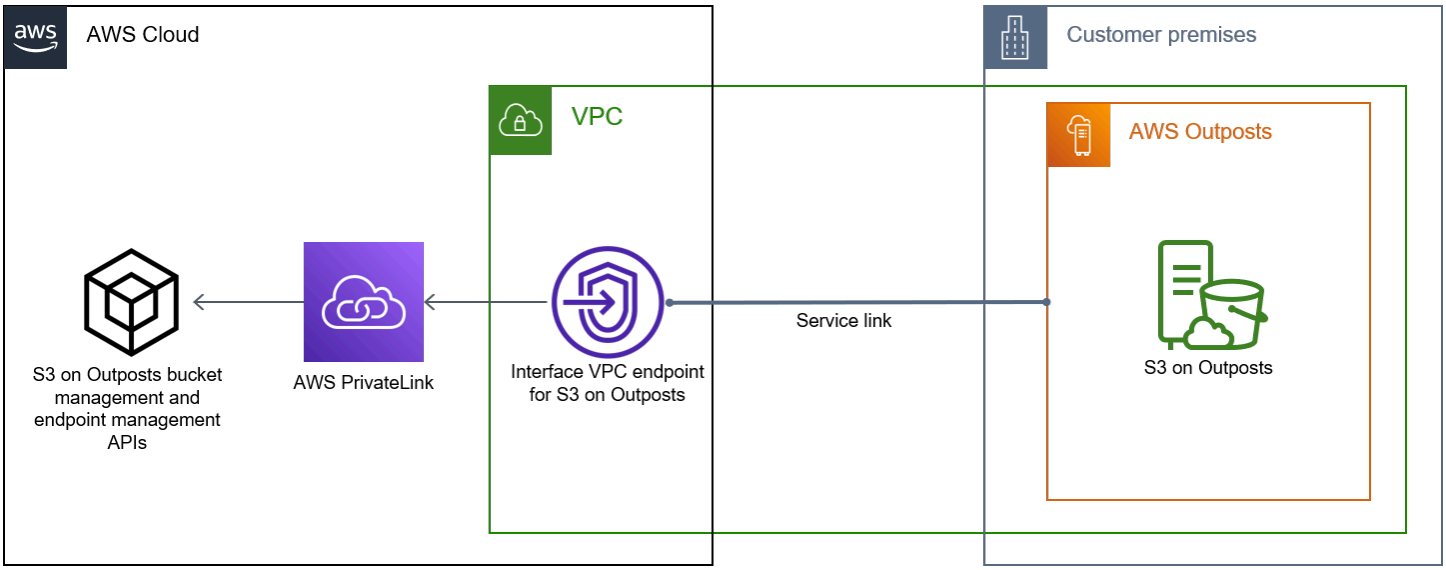
S3 en Outposts admite AWS PrivateLink, que proporciona acceso de administración directo a su almacenamiento de S3 en Outposts a través de un punto de conexión privado dentro de su red privada virtual. Esto le permite simplificar la arquitectura de su red interna y realizar operaciones de administración en el almacenamiento de objetos de Outpost mediante el uso de direcciones IP privadas en su nube privada virtual (VPC). El uso de AWS PrivateLink elimina la necesidad de utilizar direcciones IP públicas o servidores proxy.

Con AWS PrivateLink para Amazon S3 en Outposts, puede aprovisionar puntos de conexión de VPC de interfaz en la nube privada virtual (VPC) para acceder a sus API de administración de bucket y administración de puntos de conexión de S3 en Outposts. A los puntos de conexión de VPC de interfaz se puede acceder directamente desde las aplicaciones que se implementan en la VPC o en las instalaciones a través de la red privada virtual (VPN) o AWS Direct Connect. Puede acceder a las API de administración de buckets y de puntos de conexión a través de AWS PrivateLink. AWS PrivateLink no admite operaciones de API de transferencia de datos, como GET, PUT y API similares. Estas operaciones ya se transfieren de forma privada a través de la configuración de punto de acceso y punto de conexión de S3 en Outposts. Para obtener más información, consulte Redes para S3 en Outposts.

Los puntos de enlace de la interfaz se representan mediante una o más interfaces de red elásticas (elastic network interfaces, ENI) a las que se asignan direcciones IP privadas desde subredes de la VPC. Las solicitudes que se realizan a los puntos de conexión de interfaz para S3 en Outposts se enrutan automáticamente a las API de administración de buckets y de punto de conexión de S3

en Outposts en la red de AWS. Asimismo, puede acceder a los puntos de conexión de la interfaz en su VPC desde aplicaciones en las instalaciones a través de AWS Direct Connect oAWS Virtual Private Network (AWS VPN). Para obtener más información sobre cómo conectar la VPC a la red en las instalaciones, consulte la Guía del usuario de AWS Direct Connect y la Guía del usuario de AWS Site-to-Site VPN.

Los puntos de conexión de la interfaz enrutan solicitudes para las API de administración de buckets y de puntos de conexión de S3 en Outposts a través de la red de AWS y a través de AWS PrivateLink, como se ilustra en el siguiente diagrama.



Para obtener más información sobre los puntos de enlace de la interfaz, consulte Puntos de enlace de la VPC de la interfaz (AWS PrivateLink) en la Guía de AWS PrivateLink.

Temas

- Restricciones y limitaciones
- Acceso a los puntos de conexión de la interfaz de S3 en Outposts
- Actualización de una configuración DNS en las instalaciones
- Creación de un punto de conexión de VPC para S3 en Outposts
- Creación de políticas de bucket y políticas de punto de conexión de VPC para S3 en Outposts

Restricciones y limitaciones

Cuando accede a las API de administración de buckets y de puntos de conexión de S3 en Outposts a través de AWS PrivateLink, la VPC tiene una serie de limitaciones. Para obtener más

información, consulte Propiedades y limitaciones de los puntos de enlace de interfaz y Cuotas de AWS PrivateLink en la Guía de AWS PrivateLink.

Además, AWS PrivateLink no admite lo siguiente:

- Puntos de conexión del estándar federal de procesamiento de información (FIPS)
- API de transferencia de datos de S3 en Outposts, por ejemplo, operaciones de API de objetos GET, PUT y similares.
- DNS privado

Acceso a los puntos de conexión de la interfaz de S3 en Outposts

Para acceder a las API de administración de buckets y de puntos de conexión de S3 en Outposts mediante AWS PrivateLink, debe actualizar las aplicaciones para utilizar nombres de DNS específicos de cada punto de conexión. Cuando se crea un punto de conexión de interfaz, AWS PrivateLink genera dos tipos de nombres de S3 en Outposts específicos del punto de conexión: regional y zonal.

- Nombres DNS regionales: incluyen un ID único de punto de conexión de VPC, un identificador de servicio, la Región de AWS y `vpce.amazonaws.com`, por ejemplo, *`vpce-1a2b3c4d-5e6f`*`.s3-outposts.us-east-1.vpce.amazonaws.com`.
- Nombres DNS zonales: incluya un ID de punto de conexión de VPC único, la zona de disponibilidad, un identificador de servicio, Región de AWS y `vpce.amazonaws.com`, por ejemplo, *`vpce-1a2b3c4d-5e6f`*`-`**`us-east-1a`**`.s3-outposts.us-east-1.vpce.amazonaws.com`. Puede utilizar esta opción si la arquitectura aísla Zonas de disponibilidad. Por ejemplo, podría usar nombres DNS zonales para la contención de errores o para reducir los costos de transferencia de datos regionales.

> **Important**
>
> Los puntos de conexión de la interfaz de S3 en Outposts se resuelven desde el dominio de DNS público. S3 en Outposts no admite DNS privados. Utilice el parámetro `--endpoint-url` para todas las API de administración de buckets y puntos de conexión.

Ejemplos de AWS CLI

Use los parámetros `--region` y `--endpoint-url` para acceder a las API de administración de bucket y administración de punto de conexión a través de puntos de conexión de interfaz de S3 en Outposts.

Example : Utilice la URL del punto de conexión para mostrar buckets con la API de control S3

En el siguiente ejemplo, sustituya la región *us-east-1*, la URL de punto de conexión de VPC de *vpce-1a2b3c4d-5e6f.s3.us-east-1.vpce.amazonaws.com* y el ID de cuenta *111122223333* por la información adecuada.

```
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
```

Ejemplos del AWS SDK

Actualice los SDK a la versión más reciente y configure los clientes para que utilicen una URL de punto de conexión para acceder a la API de control de S3 para puntos de conexión de interfaz de S3 en Outposts. Para obtener más información, consulte los ejemplos del SDK de AWS para AWS PrivateLink.

SDK for Python (Boto3)

Example : utilice una URL de punto de conexión para acceder a la API de control de S3

En el siguiente ejemplo, sustituya la región *us-east-1* y la URL de punto de conexión de VPC *vpce-1a2b3c4d-5e6f.s3-outposts.us-east-1.vpce.amazonaws.com* por la información adecuada.

```
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'
)
```

Para obtener más información, consulte AWS PrivateLink for Amazon S3 en la guía para desarrolladores de Boto 3.

SDK for Java 2.x

Example : utilice una URL de punto de conexión para acceder a la API de control de S3

En el siguiente ejemplo, sustituya la URL de punto de conexión de VPC *vpce-1a2b3c4d-5e6f.s3-outposts.us-east-1.vpce.amazonaws.com* y la región *Region.US_EAST_1* por la información adecuada.

```
// 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()
```

Para obtener más información, consulte S3ControlClient en la Referencia de la API de AWS SDK for Java.

Actualización de una configuración DNS en las instalaciones

Al utilizar nombres DNS específicos de punto de conexión para acceder a los puntos de conexión de la interfaz de las API de administración de bucket y administración de punto de conexión de S3 en Outposts, no es necesario actualizar la resolución DNS local. Puede resolver el nombre DNS específico del punto de conexión con la dirección IP privada del punto de conexión de la interfaz desde el dominio DNS público de S3 en Outposts.

Creación de un punto de conexión de VPC para S3 en Outposts

Para crear un punto de conexión de interfaz de VPC para S3 en Outposts, consulte Crear un punto de conexión de VPC en la Guía de AWS PrivateLink.

Creación de políticas de bucket y políticas de punto de conexión de VPC para S3 en Outposts

Puede asociar una política de punto de conexión con el punto de conexión de VPC que controla el acceso a S3 en Outposts. También puede utilizar la condición `aws:sourceVpce` en las políticas de bucket de S3 en Outposts para restringir el acceso a buckets específicos desde un punto de conexión de VPC específico. Con las políticas de punto de conexión de VPC, puede controlar el

acceso a las API de administración de bucket y las API de administración de punto de conexión de S3 en Outposts. Con las políticas de bucket, puede controlar el acceso a las API de administración de bucket de S3 en Outposts. Sin embargo, no puede administrar el acceso a las acciones de objeto para S3 en Outposts mediante `aws:sourceVpce`.

Las políticas de acceso para S3 en Outposts especifican la siguiente información:

- La entidad principal de AWS Identity and Access Management (IAM) para la que se permiten o deniegan acciones.
- Las acciones de control de S3 permitidas o denegadas.
- Los recursos de S3 en Outposts en los cuales se permiten o deniegan acciones.

En los siguientes ejemplos se muestran políticas que restringen el acceso a un bucket o a un punto de conexión. Para obtener más información acerca de la conectividad de VPC, consulte Opciones de conectividad de red a VPC en el documento técnico de AWSOpciones de conectividad de Amazon Virtual Private Cloud.

> **Important**
>
> - Al aplicar las políticas de ejemplo de puntos de conexión de VPC descritos en esta sección, es posible que bloquee el acceso al bucket sin querer. Los permisos de bucket que limitan el acceso del bucket a las conexiones procedentes del punto de conexión de VPC pueden bloquear todas las conexiones al bucket. Para obtener información acerca de cómo corregir este problema, consulte Mi política de bucket tiene una VPC o un ID de punto de conexión de la VPC incorrectos. ¿Cómo puedo corregir la política de modo que pueda tener acceso al bucket? en el Centro de conocimientos de AWS Support.
> - Antes de utilizar las siguientes políticas de bucket de ejemplo, sustituya el ID del punto de conexión de VPC por un valor adecuado para su caso de uso. De lo contrario, no podrá acceder a su bucket.
> - Si la política solo permite acceder a un bucket de S3 en Outposts desde un punto de conexión de VPC específico, desactiva el acceso a la consola para ese bucket porque las solicitudes de consola no se originan en el punto de conexión de VPC especificado.

Temas

- Ejemplo: restringir el acceso a un bucket específico desde un punto de conexión de la VPC

- Ejemplo: Denegación de acceso desde un punto de conexión de VPC específico en una política de bucket de S3 en Outposts

Ejemplo: restringir el acceso a un bucket específico desde un punto de conexión de la VPC

Puede crear una política de punto de conexión que restrinja el acceso solo a buckets específicos de S3 en Outposts. La siguiente política restringe el acceso de la acción GetBucketPolicy solo a *example-outpost-bucket*. Para utilizar esta política, sustituya los valores de ejemplo por los suyos.

```
{
  "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"
    }
  ]
}
```

Ejemplo: Denegación de acceso desde un punto de conexión de VPC específico en una política de bucket de S3 en Outposts

La siguiente política de bucket de S3 en Outposts niega el acceso a GetBucketPolicy en el bucket de *example-outpost-bucket* a través del punto de conexión de VPC *vpce-1a2b3c4d*.

La condición `aws:sourceVpce` especifica el punto de conexión y no requiere un nombre de recurso de Amazon (ARN) para el recurso de punto de conexión de VPC, solo el ID de punto de conexión. Para utilizar esta política, sustituya los valores de ejemplo por los suyos.

```
{
    "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"}
            }
        }
    ]
}
```

Claves de política de autenticación de AWS Signature Version 4 (SigV4)

En la siguiente tabla se muestran las claves de condición relacionadas con la autenticación de AWS Signature Version 4 (SigV4) que puede utilizar con Amazon S3 en Outposts. En una política de bucket, puede agregar estas condiciones para imponer un comportamiento específico cuando las solicitudes se autentican mediante Signature Version 4. Para ver ejemplos de políticas, consulte Ejemplos de políticas de bucket que utilizan claves de condición relacionadas con Signature Version 4. Para obtener más información sobre las solicitudes de autenticación con Signature Version 4, consulte Autenticación de solicitudes (AWS Signature Version 4) en la Referencia de la API de Amazon Simple Storage Service

Claves aplicables para acciones de **`s3-outposts:*`** o cualquiera de las acciones de S3 en Outposts

Claves aplicables	Descripción
`s3-outposts:authType`	S3 en Outposts admite varios métodos de autenticación. Para restringir las solicitudes entrantes para que usen un método de autenticación específico, puede usar esta clave de condición opcional. Por ejemplo, puede utilizar esta clave de condición para permitir solo el encabezado `Authorization` de HTTP que se utilizará en la autenticación de solicitudes. Valores válidos: `REST-HEADER`

Claves aplicables	Descripción
	`REST-QUERY-STRING`
`s3-outposts:signatureAge`	El periodo, en milisegundos, que una firma es válida en una solicitud autenticada. Esta condición solo funciona para las direcciones URL prefirmadas. En Signature Version 4, la clave de firma es válida por un plazo máximo de siete días. Por lo tanto, las firmas también son válidas por un plazo máximo de siete días. Para obtener más información, consulte Introducción a la firma de solicitudes en la Referencia de la API de Amazon Simple Storage Service. Puede usar esta condición para limitar aún más la antigüedad de la firma. Ejemplo de valor: `600000`

Claves aplicables	Descripción
`s3-outposts:x-amz-content-sha256`	Puede utilizar esta clave de condición para no permitir el contenido sin firmar en su bucket. Cuando se utiliza Signature Version 4, para las solicitudes que utilizan el encabezado `Authorization` , agregue el encabezado `x-amz-content-sha256` en el cálculo de la firma y luego establezca su valor en la carga de hash. Puede usar esta clave de condición en su política de bucket para denegar cualquier carga en la que las cargas no estén firmadas. Por ejemplo: • Denegar las cargas que usen el encabezado `Authorization` para autenticar las solicitudes, pero no firmar la carga. Para obtener más información, consulte Transferencia de carga en un solo fragmento en la Referencia de la API de Amazon Simple Storage Service. • Denegar las cargas que utilicen URL prefirmadas. Las URL prefirmadas siempre tienen una `UNSIGNED_PAYLOAD` . Para obtener más información, consulte Autenticación de solicitudes y Métodos de autenticación en la Referencia de la API de Amazon Simple Storage Service. Valor válido: `UNSIGNED-PAYLOAD`

Ejemplos de políticas de bucket que utilizan claves de condición relacionadas con Signature Version 4

Para utilizar los siguientes ejemplos, reemplace los *`user input placeholders`* con su propia información.

Example : **`s3-outposts:signatureAge`**

La siguiente política de bucket deniega cualquier solicitud de URL prefirmada de S3 en Outposts en objetos en `example-outpost-bucket` si la firma tiene más de 10 minutos.

```
{
    "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 siguiente política de bucket solo permite las solicitudes que utilizan el encabezado `Authorization` para solicitar autenticación. Se denegará cualquier solicitud de URL prefirmada, ya que las URL prefirmadas utilizan parámetros de consulta para proporcionar información de solicitud y autenticación. Para obtener más información, consulte Métodos de autenticación en la referencia de la API de 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 siguiente política de bucket deniega cualquier carga con cargas sin firmar, como las cargas que utilizan URL prefirmadas. Para obtener más información, consulte Autenticación de solicitudes y Métodos de autenticación en la Referencia de la API de 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"
                     }
               }
         }
   ]
}
```

Políticas administradas de AWS para Amazon S3 en Outposts

Una política administrada de AWS es una política independiente que AWS crea y administra. Las políticas administradas de AWS se diseñan para ofrecer permisos para muchos casos de uso comunes, por lo que puede empezar a asignar permisos a los usuarios, grupos y roles.

Tenga presente que es posible que las políticas administradas de AWS no concedan permisos de privilegio mínimo para los casos de uso concretos, ya que están disponibles para que las utilicen

todos los clientes de AWS. Se recomienda definir políticas administradas por el cliente para los casos de uso a fin de reducir aún más los permisos.

No puede cambiar los permisos definidos en las políticas administradas por AWS. Si AWS actualiza los permisos definidos en un política administrada de AWS, la actualización afecta a todas las identidades de entidades principales (usuarios, grupos y roles) a las que está adjunta la política. Lo más probable es que AWS actualice una política administrada de AWS cuando se lance un nuevo Servicio de AWS o las operaciones de la API nuevas estén disponibles para los servicios existentes.

Para obtener más información, consulte Políticas administradas de AWS en la Guía del usuario de IAM.

Política administrada de AWS: AWSS3OnOutpostsServiceRolePolicy

Le ayuda a administrar los recursos de la red como parte del rol vinculado al servicio `AWSServiceRoleForS3OnOutposts`.

Para consultar los permisos de esta política, consulte AWSS3OnOutpostsServiceRolePolicy.

Actualizaciones de S3 en Outposts en las políticas administradas por AWS

Consulte los detalles sobre las actualizaciones de las políticas administradas por AWS para S3 en Outposts debido a que este servicio comenzó a realizar el seguimiento de estos cambios.

Cambio	Descripción	Fecha
S3 en Outposts agregado a `AWSS3OnOutpostsServiceRolePolicy`	S3 en Outposts agregado a `AWSS3OnOutpostsServiceRolePolicy` como parte de un rol vinculado a un servicio `AWSServiceRoleForS3OnOutposts`, que le ayuda a administrar los recursos de red.	3 de octubre de 2023
S3 en Outposts empezó a realizar un seguimiento de los cambios	S3 en Outposts comenzó un seguimiento de los cambios	3 de octubre de 2023

Cambio	Descripción	Fecha
	en las políticas administradas de AWS.	

Uso de roles vinculados a servicios para Amazon S3 en Outposts

Amazon S3 en Outposts utiliza roles vinculados a servicios de AWS Identity and Access Management (IAM). Un rol vinculado a un servicio es un tipo único de rol de IAM que está vinculado directamente a S3 en Outposts. Los roles vinculados a servicios los predefine S3 en Outposts e incluyen todos los permisos que el servicio necesita para llamar a otros servicios de AWS en su nombre.

Un rol vinculado a un servicio simplifica la configuración de S3 en Outposts porque ya no tendrá que agregar manualmente los permisos necesarios. S3 en Outposts define los permisos de sus roles vinculados a servicios y, a menos que esté definido de otra manera, solo S3 en Outposts puede asumir sus roles. Los permisos definidos incluyen las políticas de confianza y de permisos y que la política de permisos no se pueda adjuntar a ninguna otra entidad de IAM.

Solo puede eliminar una función vinculada a un servicio después de eliminar sus recursos relacionados. De esta forma, se protegen los recursos de S3 en Outposts, ya que se evita que se puedan eliminar accidentalmente permisos de acceso a los recursos.

Para obtener información sobre otros servicios que admiten roles vinculados a servicios, consulte Servicios de AWS que funcionan con IAM y busque los servicios que muestran Yes (Sí) en la columna Service-linked roles (Roles vinculados a servicios). Seleccione una opción Sí con un enlace para ver la documentación acerca del rol vinculado al servicio en cuestión.

Permisos de roles vinculados a servicios para S3 en Outposts

S3 en Outposts usa el rol vinculado a un servicio denominado AWSServiceRoleForS3OnOutposts para ayudarle a administrar los recursos de la red.

El rol vinculado a servicios `AWSServiceRoleForS3OnOutposts` confía en los siguientes servicios para asumir el rol:

- `s3-outposts.amazonaws.com`

La política de permisos de roles llamada `AWSS3OnOutpostsServiceRolePolicy` permite que S3 en Outposts complete las siguientes acciones en los recursos especificados:

```
{
    "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"
        }
    ]
}
```

Debe configurar permisos para permitir a una entidad de IAM (como un rol) crear, editar o eliminar un rol vinculado a servicios. Para obtener más información, consulte Permisos de roles vinculados a servicios en la Guía del usuario de IAM.

Creación de un rol vinculado a un servicio para S3 en Outposts

No necesita crear manualmente un rol vinculado a servicios. Cuando crea un punto de conexión de S3 en Outposts en la AWS Management Console, la AWS CLI o la API de AWS, S3 en Outposts crea un rol vinculado a un servicio para usted.

Si elimina este rol vinculado al servicio y necesita crearlo de nuevo, puede utilizar el mismo proceso para volver a crear el rol en su cuenta. Cuando crea un punto de conexión de S3 en Outposts, S3 en Outposts crea el rol vinculado a un servicio para usted de nuevo.

También puede utilizar la consola de IAM para crear un rol vinculado a un servicio con el caso de uso de S3 en Outposts. En la AWS CLI o la API de AWS, cree un rol vinculado al servicio con el nombre de servicio `s3-outposts.amazonaws.com`. Para obtener más información, consulte Creación de

un rol vinculado a un servicio en la Guía del usuario de IAM. Si elimina este rol vinculado al servicio, puede utilizar este mismo proceso para volver a crear el rol.

Edición de un rol vinculado a un servicio para S3 en Outposts

S3 en Outposts no le permite editar el rol vinculado a servicios `AWSServiceRoleForS3OnOutposts`. Esto incluye el nombre del rol porque es posible que varias entidades hagan referencia a él. Sin embargo, puede editar la descripción del rol mediante IAM. Para obtener más información, consulte Editar un rol vinculado a servicios en la Guía del usuario de IAM.

Eliminación de un rol vinculado a un servicio para S3 en Outposts

Si ya no necesita utilizar una característica o servicio que requiere un rol vinculado a un servicio, recomendamos que elimine dicho rol. De esta forma no tiene una entidad no utilizada que no se monitorice ni mantenga de forma activa. Sin embargo, debe limpiar los recursos del rol vinculado al servicio antes de eliminarlo manualmente.

> **Note**
>
> Si el servicio de S3 en Outposts utiliza el rol cuando intenta eliminar los recursos, es posible que la eliminación produzca un error. En tal caso, espere unos minutos e intente de nuevo la operación.

Para eliminar los recursos de S3 en Outposts utilizados por el rol AWSServiceRoleForS3OnOutposts

1. Elimine los puntos de conexión de S3 en Outposts de la Cuenta de AWS en todas las Regiones de AWS.
2. Elimine el rol vinculado a servicios con IAM.

   Utilice la consola de IAM, la AWS CLI o la API de AWS para eliminar el rol vinculado a servicios de `AWSServiceRoleForS3OnOutposts`. Para obtener más información, consulte Eliminar un rol vinculado a un servicio en la Guía del usuario de IAM.

Regiones admitidas para roles vinculados a servicios de S3 en Outposts

S3 en Outposts admite el uso de roles vinculados a servicios en todas las Regiones de AWS en las que el servicio esté disponible. Para obtener más información, consulte Regiones y puntos de conexión de S3 en Outposts.

Administración de almacenamiento de S3 en Outposts

Con Amazon S3 en Outposts, puede crear buckets de S3 en Outposts de AWS y almacenar y recuperar fácilmente objetos en las instalaciones para las aplicaciones que requieren acceso local a los datos, procesamiento local de los datos y residencia de los datos. S3 en Outposts proporciona una nueva clase de almacenamiento, S3 Outposts (OUTPOSTS), que utiliza las API de Amazon S3 y está diseñada para almacenar datos de manera duradera y redundante en múltiples dispositivos y servidores de AWS Outposts. Usted se comunica con su bucket de Outpost mediante un punto de acceso y una conexión de punto de conexión a través de una nube privada virtual (VPC). Puede usar las mismas API y características en los buckets de Outposts que en buckets de Amazon S3, como políticas de acceso, cifrado y etiquetado. Puede utilizar S3 en Outposts a través de la AWS Management Console, AWS Command Line Interface (AWS CLI), AWS SDK o la API de REST. Para obtener más información, consultar ¿Qué es Amazon S3 en Outposts?

Para obtener más información sobre cómo administrar y compartir la capacidad de almacenamiento de Amazon S3 en Outposts, consulte los siguientes temas.

Temas

Administración de control de versiones de S3 para su bucket de S3 en Outposts

Cuando está habilitado, el control de versiones de S3 guarda diversas copias de un objeto en el mismo bucket. Puede utilizar el control de versiones de S3 para conservar, recuperar y restaurar todas las versiones de los objetos almacenados en su bucket de Outposts. EL control de versiones de S3 ayuda a recuperarse de acciones no deseadas del usuario y de errores de la aplicación.

Los buckets de Amazon S3 en Outposts tienen tres estados de versiones:

- Unversioned (Sin control de versiones): si nunca ha habilitado o suspendido el control de versiones de S3 en su bucket, no tiene versiones y no muestra ningún estado de control de versiones de S3. Para obtener más información sobre el control de versiones de S3, consulte Usar el control de versiones en buckets de S3.
- Enabled (Habilitado): habilita el control de versiones de S3 para los objetos del bucket. Todos los objetos añadidos al bucket reciben un ID de versión único. Los objetos que ya existían en el bucket en el momento en que habilita el control de versiones tienen un ID de versión de `null`. Si modifica estos objetos (o cualquier otro) con otras operaciones, como PutObject, los objetos nuevos obtienen un ID de versión único.
- Suspended (Suspendido): suspende el control de versiones de S3 para los objetos del bucket. Todos los objetos añadidos al bucket tras la suspensión del control de versiones reciben el ID de versión `null`. Para obtener más información, consulte Agregar objetos a buckets con control de versiones suspendido.

Después de habilitar el control de versiones de S3 para un bucket de S3 en Outposts, nunca puede volver a un estado sin versiones. Sin embargo, puede suspender el control de versiones. Para obtener más información sobre el control de versiones de S3, consulte Usar el control de versiones en buckets de S3.

Para cada objeto de su bucket, tiene una versión actual y cero o más versiones no actuales. Para reducir los costes de almacenamiento, puede configurar las reglas del ciclo de vida de su bucket S3 para que caduquen las versiones no actuales después de un período de tiempo específico. Para obtener más información, consulte Creación y administración de una configuración de ciclo de vida para un bucket de Amazon S3 en Outposts.

En los siguientes ejemplos se muestra cómo habilitar o suspender el control de versiones para un bucket S3 on Outposts existente utilizando la AWS Management Console y la AWS Command Line Interface (AWS CLI). Para crear un bucket con el control de versiones de S3 activado, consulte Creación de un bucket de S3 en Outposts.

> **Note**
>
> La Cuenta de AWS que crea el bucket es su propietaria y la única que puede confirmarle acciones. Los buckets tienen propiedades de configuración como Outpost, etiquetas, cifrado predeterminado y valores de puntos de acceso. La configuración de punto de acceso incluye la VPC (nube virtual privada) y la política de punto de acceso para acceder a los objetos del

bucket y otros metadatos. Para obtener más información, consulte Especificaciones de S3 en Outposts.

Uso de la consola de S3

Para editar la configuración de control de versiones de S3 para su bucket

1. Inicie sesión en la AWS Management Console y abra la consola de Amazon S3 en https://console.aws.amazon.com/s3/.
2. En el panel de navegación de la izquierda, elija Outposts buckets (Buckets de Outposts).
3. Elija el bucket de Outposts para el que desea habilitar el control de versiones de S3.
4. Elija la pestaña Properties (Propiedades).
5. En Bucket Versioning (Versiones del bucket), elija Edit (Editar).
6. Edite la configuración del control de versiones de S3 para el bucket, eligiendo una de las siguientes opciones:
   - Para suspender el control de versiones de S3 y detener la creación de nuevas versiones de objetos, elija Suspend (Suspender).
   - Para habilitar el control de versiones de S3 y guardar varias copias distintas de cada objeto, elija Enable (Habilitar).
7. Elija Save changes (Guardar cambios).

Mediante AWS CLI

Para habilitar o suspender el control de versiones de S3 para su bucket mediante la AWS CLI, utilice el comando `put-bucket-versioning`, como se muestra en los siguientes ejemplos. Para utilizar estos ejemplos, sustituya *`user input placeholder`* por su propia información.

Para obtener más información, consulte put-bucket-versioning en la AWS CLIReferencia de .

Example : para habilitar el control de versiones de 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 : para suspender el control de versiones de 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
```

Creación y administración de una configuración de ciclo de vida para un bucket de Amazon S3 en Outposts

Puede usar el ciclo de vida de S3 para optimizar la capacidad de almacenamiento para Amazon S3 en Outposts. Puede crear reglas de ciclo de vida para hacer vencer los objetos a medida que envejecen o se sustituyan por versiones más recientes. Puede crear, habilitar, desactivar o eliminar una regla de ciclo de vida.

Para obtener más información acerca de S3 Lifecycle, consulte Administración del ciclo de vida del almacenamiento.

> **Note**
>
> La Cuenta de AWS que crea el bucket es su propietaria y la única que puede crear, habilitar, desactivar o eliminar una regla de ciclo de vida.

Para crear y administrar la configuración del ciclo de vida del bucket de S3 en Outposts, consulte los siguientes temas.

Temas

- Creación y administración de una regla de ciclo de vida con la AWS Management Console
- Creación y administración de una configuración de ciclo de vida mediante la AWS CLI y el SDK para Java

Creación y administración de una regla de ciclo de vida con la AWS Management Console

Puede usar el ciclo de vida de S3 para optimizar la capacidad de almacenamiento para Amazon S3 en Outposts. Puede crear reglas de ciclo de vida para hacer vencer los objetos a medida que envejecen o se sustituyan por versiones más recientes. Puede crear, habilitar, desactivar o eliminar una regla de ciclo de vida.

Para obtener más información acerca de S3 Lifecycle, consulte Administración del ciclo de vida del almacenamiento.

> **Note**
>
> La Cuenta de AWS que crea el bucket es su propietaria y la única que puede crear, habilitar, desactivar o eliminar una regla de ciclo de vida.

Para crear y administrar una regla de ciclo de vida para un S3 en Outposts utilizando la AWS Management Console, consulte los siguientes temas.

Temas

- Creación de una regla de ciclo de vida
- Habilitar una regla de ciclo de vida
- Edición de una regla de ciclo de vida
- Eliminación de una regla de ciclo de vida

Creación de una regla de ciclo de vida

1. Inicie sesión en la AWS Management Console y abra la consola de Amazon S3 en https://console.aws.amazon.com/s3/.
2. En el panel de navegación izquierdo, elija Outposts buckets (Buckets de Outposts).
3. Elija el bucket de Outposts para el que desea crear una regla del ciclo de vida.
4. Seleccione la pestaña Management (Administración) y seleccione Create Lifecycle rule (Crear regla de ciclo de vida).
5. Introduzca un valor para Lifecycle rule name (Nombre de regla de ciclo de vida).
6. En Rule scope (Ámbito de rol), elija una de las siguientes opciones:
   - Para limitar el alcance para filtros específicos, elija Limit the scope of this rule using one or more filters (Limitar el alcance de esta regla con uno o más filtros). A continuación, agregue un filtro de prefijo, etiquetas o tamaño del objeto.
   - Para aplicar la regla a todos los objetos del bucket, elija Apply to all objects in the bucket (Aplicar a todos los objetos del bucket).
7. En Lifecycle rule actions (Acciones de regla de ciclo de vida), elija una de las siguientes opciones:

- Expire current versions of objects (Expirar las versiones actuales de objetos): para los buckets habilitados para el control de versiones, S3 en Outposts agrega un marcador de eliminación y retiene los objetos como versiones no actuales. Para los buckets que no utilizan el control de versiones de S3, S3 en Outposts elimina permanentemente los objetos.
- Permanently delete noncurrent versions of objects (Eliminar permanentemente las versiones no actuales de los objetos): S3 en Outposts elimina permanentemente las versiones no actuales de los objetos.
- Delete expired object delete markers or incomplete multipart uploads (Eliminar marcadores de eliminación de objetos vencidos o cargas multiparte incompletas): S3 en Outposts elimina permanentemente los marcadores de eliminación de objetos vencidos o cargas multiparte incompletas.

  Si limita el ámbito de su regla de ciclo de vida mediante etiquetas de objetos, no puede elegir Delete expired object delete markers (Eliminar marcadores de eliminación de objetos caducados). Tampoco puede elegir Delete expired object delete markers (Eliminar marcadores de eliminación de objetos caducados) si elige Expire current object versions (Expirar las versiones actuales de objetos).

> **Note**
>
> Los filtros basados en el tamaño no se pueden usar con marcadores de eliminación ni con cargas multiparte incompletas.

8. Si seleccionó Expire current versions of objects (Expirar las versiones actuales de los objetos) o Permanently delete noncurrent versions of objects (Eliminar permanentemente las versiones no actuales de los objetos), configure el desencadenador de la regla en función de una fecha específica o de la antigüedad del objeto.
9. Si eligió Delete expired object delete markers (Eliminar marcadores de eliminación de objetos caducados), para confirmar que desea eliminar los marcadores de eliminación de objetos caducados, seleccione Delete expired object delete markers (Eliminar marcadores de eliminación de objetos caducados).
10. En Timeline Summary (Resumen de línea temporal), revise su regla de ciclo de vida y seleccione Create rule (Crear regla).

Habilitar una regla de ciclo de vida

Para habilitar o deshabilitar una regla del ciclo de vida del bucket

1. Abra la consola de Amazon S3 en https://console.aws.amazon.com/s3.
2. En el panel de navegación de la izquierda, elija Outposts buckets (Buckets de Outposts).
3. Elija el bucket de Outposts para el que desea habilitar o deshabilitar una regla del ciclo de vida.
4. Elija la pestaña Management (Administración) y a continuación en Lifecycle rule (Regla de ciclo de vida), elija la regla que desea habilitar o deshabilitar.
5. En Action (Acción), elija Enable or disable rule (Habilitar o deshabilitar regla).

Edición de una regla de ciclo de vida

1. Abra la consola de Amazon S3 en https://console.aws.amazon.com/s3.
2. En el panel de navegación de la izquierda, elija Outposts buckets (Buckets de Outposts).
3. Elija el bucket de Outposts para el que desea editar una regla del ciclo de vida.
4. Elija la pestaña Management (Administración) y elija la regla del ciclo de vida que desea editar.
5. (opcional) Actualice el valor de Lifecycle rule name (Nombre de la regla del ciclo de vida).
6. En Rule scope (Ámbito de regla), modifique el ámbito según sea necesario:
   - Para limitar el alcance para filtros específicos, elija Limit the scope of this rule using one or more filters (Limitar el alcance de esta regla con uno o más filtros). A continuación, agregue un filtro de prefijo, etiquetas o tamaño del objeto.
   - Para aplicar la regla a todos los objetos del bucket, elija Apply to all objects in the bucket (Aplicar a todos los objetos del bucket).
7. En Lifecycle rule actions (Acciones de regla de ciclo de vida), elija una de las siguientes opciones:
   - Expire current versions of objects (Expirar las versiones actuales de objetos): para los buckets habilitados para el control de versiones, S3 en Outposts agrega un marcador de eliminación y retiene los objetos como versiones no actuales. Para los buckets que no utilizan el control de versiones de S3, S3 en Outposts elimina permanentemente los objetos.
   - Permanently delete noncurrent versions of objects (Eliminar permanentemente las versiones no actuales de los objetos): S3 en Outposts elimina permanentemente las versiones no actuales de los objetos.

- Delete expired object delete markers or incomplete multipart uploads (Eliminar marcadores de eliminación de objetos vencidos o cargas multiparte incompletas): S3 en Outposts elimina permanentemente los marcadores de eliminación de objetos vencidos o cargas multiparte incompletas.

  Si limita el ámbito de su regla de ciclo de vida mediante etiquetas de objetos, no puede elegir Delete expired object delete markers (Eliminar marcadores de eliminación de objetos caducados). Tampoco puede elegir Delete expired object delete markers (Eliminar marcadores de eliminación de objetos caducados) si elige Expire current object versions (Expirar las versiones actuales de objetos).

> **Note**
>
> Los filtros basados en el tamaño no se pueden usar con marcadores de eliminación ni con cargas multiparte incompletas.

8. Si seleccionó Expirar las versiones actuales de los objetos o Eliminar permanentemente las versiones no actuales de los objetos, configure el desencadenador de la regla en función de una fecha específica o de la antigüedad del objeto.
9. Si eligió Delete expired object delete markers (Eliminar marcadores de eliminación de objetos caducados), para confirmar que desea eliminar los marcadores de eliminación de objetos caducados, seleccione Delete expired object delete markers (Eliminar marcadores de eliminación de objetos caducados).
10. Seleccione Save (Guardar).

Eliminación de una regla de ciclo de vida

1. Abra la consola de Amazon S3 en https://console.aws.amazon.com/s3.
2. En el panel de navegación de la izquierda, elija Outposts buckets (Buckets de Outposts).
3. Elija el bucket de Outposts para el que desea eliminar una regla de estilo de vida.
4. Elija la pestaña Management (Administración) y luego en Lifecycle rule (Regla de ciclo de vida), elija la regla que desea eliminar.
5. Elija Eliminar.

Creación y administración de una configuración de ciclo de vida mediante la AWS CLI y el SDK para Java

Puede usar el ciclo de vida de S3 para optimizar la capacidad de almacenamiento para Amazon S3 en Outposts. Puede crear reglas de ciclo de vida para hacer vencer los objetos a medida que envejecen o se sustituyan por versiones más recientes. Puede crear, habilitar, desactivar o eliminar una regla de ciclo de vida.

Para obtener más información acerca de S3 Lifecycle, consulte Administración del ciclo de vida del almacenamiento.

> **Note**
>
> La Cuenta de AWS que crea el bucket es su propietaria y la única que puede crear, habilitar, desactivar o eliminar una regla de ciclo de vida.

Para crear y administrar una configuración de ciclo de vida para un bucket de S3 en Outposts mediante AWS Command Line Interface (AWS CLI) y AWS SDK for Java, consulte los siguientes ejemplos.

Temas

- Colocación de una configuración del ciclo de vida
- Obtención de la configuración de ciclo de vida en un bucket de S3 en Outposts

Colocación de una configuración del ciclo de vida

AWS CLI

En el siguiente ejemplo de AWS CLI, se aplica una política de configuración del ciclo de vida en un bucket de Outposts. Esta política especifica que todos los objetos que tienen el prefijo marcado (*myprefix*) y las etiquetas vencen después de 10 días. Para utilizar este ejemplo, sustituya *user input placeholder* por su propia información.

1. Guarde la política de configuración del ciclo de vida en un archivo JSON. En este ejemplo, el archivo se denomina `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. Envíe el archivo JSON como parte del comando de la CLI `put-bucket-lifecycle-configuration`. Para usar este comando, sustituya *`user input placeholder`* por su propia información. Para obtener más información acerca de este comando, consulte [put-bucket-lifecycle-configuration] en la Referencia de AWS CLI.

```
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

En el siguiente ejemplo del SDK para Java, se aplica una configuración del ciclo de vida en un bucket de Outposts. Esta configuración de ciclo de vida especifica que todos los objetos que tienen el prefijo marcado (*`myprefix`*) y las etiquetas vencen después de 10 días. Para utilizar

este ejemplo, sustituya *user input placeholder* por su propia información. Para obtener más información, consulte PutBucketLifecycleConfiguration en la Referencia de la API de Amazon Simple Storage Service.

```
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());

}
```

Obtención de la configuración de ciclo de vida en un bucket de S3 en Outposts

AWS CLI

En el siguiente ejemplo de la AWS CLI, se obtiene una configuración del ciclo de vida en un bucket de Outposts. Para usar este comando, sustituya *user input placeholder* por su propia información. Para obtener más información acerca de este comando, consulte get-bucket-lifecycle-configuration en la Referencia de AWS CLI.

```
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

En el siguiente ejemplo del SDK para Java, se obtiene una configuración del ciclo de vida para un bucket de Outposts. Para obtener más información, consulte GetBucketLifecycleConfiguration en la Referencia de la API de Amazon Simple Storage Service.

```
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());

}
```

Replicación de objetos para S3 en Outposts

Si tiene S3 Replication activado en AWS Outposts, puede configurar Amazon S3 en Outposts para que replique automáticamente objetos de S3 en diferentes Outposts o entre buckets del mismo Outpost. Puede utilizar S3 Replication en Outposts para conservar varias réplicas de sus datos en

el mismo o en diferentes Outposts, a fin de cumplir con las necesidades de residencia de datos. S3 Replication en Outposts ayuda a satisfacer sus necesidades de almacenamiento compatibles y al intercambio de datos entre cuentas. Si necesita asegurarse de que las réplicas de los objetos sean idénticos a los datos de origen, puede usar S3 Replication en Outposts para realizar réplicas de sus objetos para conservar todos los metadatos, como la hora de creación del objeto original, las etiquetas y los ID de versión.

S3 Replication en Outposts también proporciona métricas detalladas y notificaciones para monitorear el estado de la replicación de objetos entre buckets. Puede utilizar Amazon CloudWatch para monitorear el progreso de la replicación mediante el seguimiento de los bytes pendientes de replicación, las operaciones pendientes de replicación y la latencia de replicación entre los buckets de origen y de destino. Para diagnosticar y corregir rápidamente los problemas de configuración, también puede configurar Amazon EventBridge para que reciba notificaciones sobre errores en los objetos de replicación. Para obtener más información, consulte Administración de la replicación.

Temas

Configuración de replicación

S3 en Outposts almacena una configuración de replicación como XML. En el archivo XML de configuración de reproducción, usted especifica un rol de AWS Identity and Access Management (IAM) y una o más reglas.

```
<ReplicationConfiguration>
    <Role>IAM-role-ARN</Role>
    <Rule>
        ...
    </Rule>
    <Rule>
         ...
```

```
    </Rule>
     ...
</ReplicationConfiguration>
```

S3 en Outposts no puede replicar objetos sin su permiso. Usted otorga permisos a S3 en Outposts con el rol de IAM que especifique en la configuración de la replicación. S3 en Outposts asume el rol de IAM para replicar objetos en su nombre. Debe conceder los permisos necesarios para el rol de IAM para que pueda empezar a replicar. Para obtener más información sobre estos permisos para S3 en Outposts, consulte Creación de un rol de IAM.

Usted agrega una regla en una configuración de replicación en los siguientes casos:

- Desea replicar todos los objetos.
- Desea replicar un subconjunto de objetos. Identifica el subconjunto de objetos añadiendo un filtro en la regla. En el filtro, usted especifica un prefijo de clave de objeto, etiquetas o una combinación de ambos, para identificar el subconjunto de objetos a los que se aplica la regla.

Agrega varias reglas en una configuración de replicación si desea replicar un subconjunto diferente de objetos. En cada regla, se especifica un filtro que selecciona un subconjunto diferente de objetos. Por ejemplo, puede elegir replicar objetos que tengan los prefijos de clave `tax/` o `document/`. Para ello, agregue dos reglas, una que especifique el filtro de prefijo de clave `tax/` y otro que especifique el prefijo de clave `document/`.

Para obtener más información sobre la configuración de replicación y las reglas de replicación de S3 en Outposts, consulte ReplicationConfiguration en la referencia de la API de Amazon Simple Storage Service.

Requisitos de S3 Replication en Outposts

La replicación requiere lo siguiente:

- El rango CIDR de Outpost de destino debe estar asociado a la tabla de subredes de Outpost de origen. Para obtener más información, consulte Requisitos previos para crear reglas de replicación.
- Ambos buckets de origen y destino deben tener activado el control de versiones de S3. Para obtener más información sobre el control de versiones, consulte Administración de control de versiones de S3 para su bucket de S3 en Outposts.

- Amazon S3 en Outposts debe tener permisos para replicar objetos en su nombre del bucket de origen en el bucket de destino. Esto significa que debe crear un rol de servicio para delegar los permisos `GET` y `PUT` a S3 en Outposts.

  1. Antes de crear el rol de servicio, debe tener el permiso `GET` en el bucket de origen y el permiso `PUT` en el bucket de destino.
  2. Para crear el rol de servicio para delegar los permisos a S3 en Outposts, primero debe configurar los permisos para permitir que una entidad de IAM (un usuario o un rol) realice las acciones `iam:CreateRole` y `iam:PassRole`. A continuación, permite que una entidad de IAM cree el rol de servicio. Para que S3 en Outposts asuma el rol de servicio en su nombre y delegue los permisos `GET` y `PUT` a S3 en Outposts, debe asignar las políticas de confianza y de permisos necesarias al rol. Para obtener más información sobre estos permisos para S3 en Outposts, consulte Creación de un rol de IAM. Para obtener más información sobre cómo crear un rol de servicio, consulte Creación de un rol de servicio.

¿Qué se replica?

De forma predeterminada, S3 en Outposts replica lo siguiente:

- Objetos creados después de añadir una configuración de replicación.
- Metadatos de objeto desde los objetos de origen hasta las réplicas. Para obtener información acerca de la replicación de metadatos a partir de las réplicas de los objetos de origen, consulte Estado de replicación si la sincronización de modificación de réplica de Amazon S3 en Outposts está habilitada.
- Etiquetas de objeto, si las hay.

Cómo afectan las operaciones de eliminación a la replicación

Si elimina un objeto del bucket de origen, las siguientes acciones se producen de forma predeterminada:

- Si realiza una solicitud `DELETE` sin especificar un ID de versión del objeto, S3 en Outposts añade un marcador de eliminación. S3 en Outposts se ocupa del marcador de eliminación de la siguiente manera:
  - S3 en Outposts no replica el marcador de eliminación de forma predeterminada.

  - Sin embargo, puede agregar la replicación de marcador de eliminación a reglas no basadas en etiquetas. Para obtener más información acerca de cómo habilitar la replicación del marcador de eliminación en su configuración de replicación, consulte Uso de la consola de S3.
- Si especifica un ID de versión de objeto para eliminar en una solicitud de `DELETE`, S3 en Outposts elimina esa versión del objeto en el bucket de origen de forma permanente. Sin embargo, no replica la eliminación en los buckets de destino. En otras palabras, no elimina la misma versión del objeto de los buckets de destino. Este comportamiento protege los datos de eliminaciones malintencionadas.

Elementos que no se replican

De forma predeterminada, S3 en Outposts no replica lo siguiente:

- Los objetos en el bucket de origen que son réplicas, creadas por otra regla de replicación. Por ejemplo, imagine que configura la replicación donde el bucket A es el origen y el bucket B es el destino. Ahora, supongamos que añade otra configuración de replicación donde el bucket B es el de origen y el bucket C es el de destino. En este caso, los objetos en el bucket B que son réplicas de objetos en el bucket A no se replican en el bucket C.
- Objetos en el bucket de origen que ya se han replicado en un destino diferente. Por ejemplo, si cambia el bucket de destino en una configuración de replicación existente, S3 en Outposts no replicará los objetos de nuevo.
- Objetos creados con cifrado del lado del servidor con claves de cifrado proporcionadas por los clientes (SSE-C).
- Actualiza a subrecursos de bucket.

  Por ejemplo, si cambia la configuración del ciclo de vida en una configuración de notificación al bucket de origen, estos cambios no se aplican al bucket de destino. Esta característica permite tener diferentes configuraciones en los buckets de origen y destino.
- Acciones realizadas por la configuración del ciclo de vida.

  Por ejemplo, si activa la configuración del ciclo de vida en el bucket de origen y configura acciones de vencimiento, S3 en Outposts crea marcadores de eliminación para los objetos vencidos en el bucket de origen, pero no replica esos marcadores en los buckets de destino. Si desea que se aplique la misma configuración de ciclo de vida a los buckets de origen y destino, habilite la misma configuración de ciclo de vida en ambos. Para obtener más información acerca de la configuración del ciclo de vida, consulte Administración del ciclo de vida del almacenamiento.

¿Qué no admite S3 Replication en Outposts?

Las siguientes características de S3 Replication no son compatibles actualmente con S3 en Outposts:

- Control del tiempo de replicación de S3 (S3 RTC) S3 RTC no es compatible porque el tráfico de objetos en S3 Replication en Outposts viaja a través de la red en las instalaciones (la puerta de enlace local). Para obtener más información acerca de las puerta de enlace locales, consulte el temas sobre trabajar con gateways locales en la guía del usuario de AWS Outposts.
- S3 Replication para operaciones por lotes.

Configuración de la replicación

> **Note**
>
> Los objetos que había en el bucket antes de configurar la regla de replicación no se replican automáticamente. En otras palabras, Amazon S3 en Outposts no replica los objetos retroactivamente. Para replicar objetos creados antes de que configurara la replicación, puede utilizar la operación de la API `CopyObject` para copiarlos en el mismo bucket. Una vez copiados los objetos, aparecen como objetos «nuevos» en el bucket y se les aplica la configuración de replicación. Para obtener más información sobre cómo se copia un objeto, consulte Copia de un objeto en un bucket de Amazon S3 en Outposts utilizando AWS SDK for Java y CopyObject en la Referencia de la API de Amazon Simple Storage Service.

Para activar la Replicación de S3 en Outposts, añada una regla de replicación a su bucket de Outposts de origen. La regla de replicación indica a S3 en Outposts que replique los objetos de la forma especificada. En la configuración de replicación, debe proporcionar lo siguiente:

- El punto de acceso al bucket de Outposts de origen: el nombre de recurso de Amazon (ARN) o el alias del punto de acceso desde el que desea que S3 en Outposts replique los objetos. Para obtener más información, consulte Uso de un alias de estilo de bucket para su punto de acceso de bucket de S3 en Outposts.
- Los objetos que desea replicar: puede replicar todos los objetos del bucket de Outposts de origen o de un subconjunto. Para identificar un subconjunto, proporcione un prefijo de nombre de clave, una o más etiquetas de objeto, o ambos en la configuración.

Por ejemplo, si configura una regla de replicación para replicar solo objetos con el prefijo de nombre de clave `Tax/`, S3 en Outposts replica objetos con claves como `Tax/doc1` o `Tax/doc2`. Pero no replica objetos con la clave `Legal/doc3`. Si especifica un prefijo y una o más etiquetas, S3 en Outposts replica solo los objetos que tienen el prefijo de clave específico y las etiquetas.

- El bucket de Outposts de destino: el ARN o el alias del punto de acceso del bucket en el que desea que S3 en Outposts replique los objetos.

Puede configurar la regla de replicación mediante la API de REST, los SDK de AWS, la AWS Command Line Interface (AWS CLI) o la consola de Amazon S3.

S3 en Outposts también proporciona operaciones de API para que admita la configuración de reglas de replicación. Para obtener más información, consulte los siguientes temas en la referencia de la API de Amazon Simple Storage Service:

- PutBucketReplication
- GetBucketReplication
- DeleteBucketReplication

Temas

- Requisitos previos para crear reglas de replicación
- Creación de reglas de replicación en Outposts

Requisitos previos para crear reglas de replicación

Temas

- Conectar las subredes de Outpost de origen y destino
- Creación de un rol de IAM

Conectar las subredes de Outpost de origen y destino

Para que el tráfico de replicación vaya de su Outpost de origen a su Outpost de destino a través de la puerta de enlace local, debe agregar una nueva ruta para configurar la red. Debe conectar entre sí los rangos de red del enrutamiento entre dominios sin clases (CIDR) de sus puntos de acceso. Para cada par de puntos de acceso, debe configurar esta conexión solo una vez.

Algunos pasos para configurar la conexión varían según el tipo de acceso de los puntos de conexión de Outposts que estén asociados a sus puntos de acceso. El tipo de acceso para los puntos de conexión es Privado (enrutamiento directo a la nube privada virtual [VPC] para AWS Outposts) o IP propiedad del cliente (un grupo de direcciones IP propiedad del cliente [grupo CoIP] dentro de la red en las instalaciones).

Paso 1: Encontrar el rango de CIDR de su punto de conexión de Outposts de origen

Para encontrar el rango de CIDR de su punto de conexión de origen asociado a su punto de acceso de origen

1. Inicie sesión en la AWS Management Console y abra la consola de Amazon S3 en https://console.aws.amazon.com/s3/.
2. En el panel de navegación izquierdo, elija Outposts buckets (Buckets de Outposts).
3. En la lista Buscar buckets de Outposts, elija el bucket de origen que desea replicar.
4. Elija la pestaña Puntos de acceso de Outposts y elija el punto de acceso de Outposts para el bucket de origen de su regla de replicación.
5. Seleccione el punto de conexión de Outposts.
6. Copie el ID de subred para usarlo en el paso 5.
7. El método que utilice para encontrar el rango de CIDR del punto de conexión de Outposts de origen depende del tipo de acceso de su punto de conexión.

   En la sección Información general sobre los puntos de enlace de Outposts, consulte Tipo de acceso.

   - Si el tipo de acceso es Privado, copie el valor del Enrutamiento entre dominios sin clases (CIDR) para usarlo en el paso 6.
   - Si el tipo de acceso es IP propiedad del cliente, haga lo siguiente:
     1. Copie el valor del grupo IPv4 propiedad del cliente para usarlo más adelante como ID del grupo de direcciones.
     2. Abra la consola de AWS Outposts en https://console.aws.amazon.com/outposts/.
     3. En el panel de navegación, elija Tablas de enrutamiento de puerta de enlace de tránsito.
     4. Elija el valor de ID de tabla de enrutamiento de puerta de enlace en las instalaciones del Outpost de origen.
     5. En el panel de detalles, elija la pestaña Grupos de CoIP. Pegue el valor de su ID de grupo de CoIP copiado anteriormente en el cuadro de búsqueda.

6. Para el grupo de CoIP coincidente, copie el valor de CIDR correspondiente de su punto de conexión de Outposts de origen para usarlo en el paso 6.

Paso 2: Buscar el ID de subred y el rango de CIDR de su punto de conexión de Outposts de destino

Para encontrar el ID de subred y el rango de CIDR de su punto de conexión de destino asociados a su punto de acceso de destino, siga los mismos subpasos del paso 1 y cambie el punto de conexión de Outposts de origen por el punto de conexión de Outposts de destino cuando aplique esos subpasos. Copie el valor del ID de subred del punto de conexión de Outposts de destino para usarlo en el paso 6. Copie el valor de CIDR del punto de conexión de Outposts de destino para usarlo en el paso 5.

Paso 3: Encontrar el ID de puerta de enlace local del Outpost de origen

Para encontrar el ID de puerta de enlace local del Outpost de origen

1. Abra la consola de AWS Outposts en https://console.aws.amazon.com/outposts/.
2. En el panel de navegación izquierdo, elija Gateways locales.
3. En la página Gateways locales, busque el ID de Outpost del Outpost de origen que quiere utilizar para la replicación.
4. Copie el valor del ID de puerta de enlace local del Outpost de origen para usarlo en el paso 5.

Para obtener información acerca de las puerta de enlaces locales, consulte el tema sobre Gateways locales en la Guía del usuario de AWS Outposts.

Paso 4: Encontrar el ID de puerta de enlace local del Outpost de destino

Para encontrar el ID de puerta de enlace local del Outpost de destino, siga los mismos subpasos del paso 3, excepto buscar el ID de Outpost del Outpost de destino. Copie el valor del ID de puerta de enlace local del Outpost de destino para usarlo en el paso 6.

Paso 5: Configurar la conexión desde la subred de Outpost de origen a la subred de Outpost de destino

Para conectarse desde la subred de Outpost de origen a la subred de Outpost de destino

1. Inicie sesión en la AWS Management Console y abra la consola de VPC en https://console.aws.amazon.com/vpc/.

2. En el panel de navegación izquierdo, elija Subnets.
3. En el cuadro de búsqueda, introduzca el ID de subred del punto de conexión de Outposts de origen que ha encontrado en el paso 1. Elija una subred con el ID de subred correspondiente.
4. Para el elemento de subred coincidente, elija el valor de Tabla de enrutamiento de esta subred.
5. En la página con una tabla de enrutamiento seleccionada, elija Acciones y, a continuación, elija Editar rutas.
6. En la pestaña Editar rutas, elija Añadir rutas.
7. En Destino, introduce el rango de CIDR del punto de conexión de Outposts de destino que encontraste en el paso 2.
8. En Objetivo, elija Gateway local de Outpost e introduzca el ID de puerta de enlace local de su Outpost de origen que ha encontrado en el paso 3.
9. Elija Save changes (Guardar cambios).
10. Asegúrese de que el Estado de la ruta sea Activo.

Paso 6: Configurar la conexión desde la subred de Outpost de destino a la subred de Outpost de origen

1. Inicie sesión en la AWS Management Console y abra la consola de VPC en https://console.aws.amazon.com/vpc/.
2. En el panel de navegación izquierdo, elija Subnets.
3. En el cuadro de búsqueda, introduzca el ID de subred del punto de conexión de Outposts de destino que ha encontrado en el paso 2. Elija una subred con el ID de subred correspondiente.
4. Para el elemento de subred coincidente, elija el valor de Tabla de enrutamiento de esta subred.
5. En la página con una tabla de enrutamiento seleccionada, elija Acciones y, a continuación, elija Editar rutas.
6. En la pestaña Editar rutas, elija Añadir rutas.
7. En Destino, introduzca el rango de CIDR del punto de conexión de Outposts de origen que ha encontrado en el paso 1.
8. En Objetivo, elija Gateway local de Outpost e introduzca el ID de puerta de enlace local del Outpost de destino que ha encontrado en el paso 4.
9. Elija Save changes (Guardar cambios).
10. Asegúrese de que el Estado de la ruta sea Activo.

Después de conectar los rangos de redes de CIDR de sus puntos de acceso de origen y destino, debe crear un rol de AWS Identity and Access Management (IAM).

Creación de un rol de IAM

De forma predeterminada, todos los recursos de S3 en Outposts (buckets, objetos y subrecursos relacionados) son privados y solo el propietario del recurso puede acceder a él. S3 en Outposts necesita permisos de lectura y replicación de objetos del bucket de Outposts de origen. Para conceder estos permisos, crea un rol de servicio de IAM y luego especifique ese rol en la configuración de replicación.

En esta sección se explica la política de confianza y la política de permisos mínimos necesarios. Los tutoriales de ejemplo proporcionan instrucciones paso a paso para crear un rol de IAM. Para obtener más información, consulte Creación de reglas de replicación en Outposts. Para obtener más información acerca de los roles de IAM, consulte Roles de IAM en Guía del usuario de IAM.

- En el siguiente ejemplo, se muestra una política de confianza donde se identifica a S3 en Outposts como la entidad principal del servicio que puede asumir el rol.

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

- En el siguiente ejemplo, se muestra una política de acceso donde se concede al rol permisos para realizar tareas de replicación en su nombre. Cuando S3 en Outposts asume el rol, adopta los permisos se hayan especificado en esta política. Para utilizar esta política, sustituya *user input placeholders* por su información. Asegúrese de sustituirlos por los ID de Outpost de sus Outposts de origen y destino y los nombres de los buckets y los puntos de acceso de sus buckets de Outposts de origen y destino.

```
{
   "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 política de acceso concede permisos para las siguientes acciones:

- `s3-outposts:GetObjectVersionForReplication`: se otorga permiso para esta acción a todos los objetos para que S3 en Outposts pueda obtener una versión de objeto específica asociada a cada objeto.

- `s3-outposts:GetObjectVersionTagging`: los permisos para esta acción en los objetos del bucket *SOURCE-OUTPOSTS-BUCKET* (el bucket de origen) permiten que S3 en Outposts lea las etiquetas de objetos para la replicación. Para obtener más información, consulte Agregar etiquetas para los buckets de S3 en Outposts. Si S3 en Outposts no tiene el permiso, replica los objetos pero no las etiquetas de objetos.

- `s3-outposts:ReplicateObject` y `s3-outposts:ReplicateDelete`: los permisos para estas acciones en todos los objetos del bucket *`DESTINATION-OUTPOSTS-BUCKET`* (el bucket de destino) autorizan a S3 en Outposts a replicar los objetos o los marcadores de eliminación en el bucket de Outposts de destino. Para obtener información acerca de los marcadores de eliminación, consulte Cómo afectan las operaciones de eliminación a la replicación.

  > **Note**
  >
  > - El permiso para la acción `s3-outposts:ReplicateObject` en el bucket *`DESTINATION-OUTPOSTS-BUCKET`* (bucket de destino) también permite la replicación de las etiquetas de objetos. Por lo tanto, no es necesario que conceda permiso de forma explícita para la acción `s3-outposts:ReplicateTags`.
  > - Para la replicación entre cuentas, el propietario del bucket de Outposts de destino debe actualizar su política de bucket para conceder permiso para la acción `s3-outposts:ReplicateObject` en el *`DESTINATION-OUTPOSTS-BUCKET`*. La acción `s3-outposts:ReplicateObject` permite a S3 en Outposts replicar los objetos y las etiquetas de objetos en el bucket de Outposts de destino.

Para obtener una lista de las acciones de S3 en Outposts, consulte Acciones definidas por Amazon S3 en Outposts.

> **Important**
>
> La Cuenta de AWS propietaria del rol de IAM debe tener los permisos para las acciones que concede al rol de IAM.
> Por ejemplo, imagine que el bucket de Outposts de origen contiene objetos que pertenecen a otra Cuenta de AWS. El propietario de los objetos debe conceder explícitamente los permisos necesarios a la Cuenta de AWS que posee el rol de IAM a través de la política de punto de acceso y de bucket. De lo contrario, S3 en Outposts no puede acceder a los objetos y no se pueden replicar los objetos.
> Los permisos aquí descritos están relacionados con la configuración de replicación mínima. Si elige agregar configuraciones de replicación opcionales, debe otorgar permisos adicionales a S3 en Outposts.

Concesión de permisos cuando los buckets de Outposts de origen y destino pertenecen a diferentes Cuentas de AWS

Cuando los buckets de Outposts de origen y destino no pertenecen a las mismas cuentas, el propietario del bucket de Outposts de destino debe actualizar las políticas de buckets y de puntos de acceso para el bucket de destino. Estas políticas deben conceder permisos al propietario del bucket de Outposts de origen y al rol de servicio de IAM para que puedan realizar acciones de replicación, tal como se muestra en los siguientes ejemplos de políticas, pues de lo contrario se producirá un error en la replicación. En estos ejemplos de política, *DESTINATION-OUTPOSTS-BUCKET* es el bucket de destino. Para utilizar estos ejemplos de política, sustituya *user input placeholders* por su información.

Si va a crear el rol de servicio de IAM de forma manual, defina la ruta del rol como `role/service-role/`, tal como se muestra en los siguientes ejemplos de políticas. Para obtener más información, consulte ARN de IAM en la guía del usuario de 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 los objetos en el bucket de Outposts de origen tienen etiquetas, tenga en cuenta lo siguiente:
> Si el propietario del bucket de Outposts de origen concede permisos a S3 en Outposts para las acciones `s3-outposts:GetObjectVersionTagging` y `s3-outposts:ReplicateTags` para replicar las etiquetas de los objetos (mediante el rol de IAM), Amazon S3 replicará las etiquetas junto con los objetos. Para obtener información acerca del rol de IAM, consulte Creación de un rol de IAM.

Creación de reglas de replicación en Outposts

La replicación de S3 en Outposts consiste en la replicación automática y asíncrona de los objetos de los buckets en la misma o en diferentes AWS Outposts. La replicación copia los objetos que se acaban de crear y las actualizaciones de objetos de un bucket de Outposts de origen a un bucket o buckets de Outposts de destino. Para obtener más información, consulte Replicación de objetos para S3 en Outposts.

> **Note**
>
> Los objetos que había en el bucket de Outposts de origen antes de que configurara las reglas de replicación no se replican. En otras palabras, S3 en Outposts no replica los objetos retroactivamente. Para replicar objetos creados antes de que configurara la replicación, puede utilizar la operación de la API `CopyObject` para copiarlos en el mismo bucket. Una vez copiados los objetos, aparecen como objetos «nuevos» en el bucket y se les aplica la configuración de replicación. Para obtener más información sobre cómo se copia un objeto, consulte Copia de un objeto en un bucket de Amazon S3 en Outposts utilizando AWS SDK for Java y CopyObject en la Referencia de la API de Amazon Simple Storage Service.

Al configurar la replicación, se agregan reglas de replicación al bucket de Outposts de origen. Las reglas de replicación definen qué objetos del bucket de Outposts de origen se deben replicar y el bucket o buckets de Outposts de destino donde se almacenarán los objetos replicados. Puede crear una regla para replicar todos los objetos en un bucket o un subconjunto de objetos con un prefijo de nombre de clave específico, una o varias etiquetas de objeto, o ambos métodos. El bucket de Outposts de destino puede estar en el mismo Outpost que el bucket de Outpost de origen o puede estar en un Outpost diferente.

Para las reglas de replicación de S3 en Outposts, debe proporcionar tanto el nombre de recurso de Amazon (ARN) del punto de acceso del bucket de Outposts de origen como el ARN del punto de acceso del bucket de Outposts de destino, en lugar de los nombres de los buckets de Outposts de origen y destino.

Si especifica un ID de versión de objeto para eliminarlo, S3 en Outposts elimina esa versión del objeto del bucket de Outposts de origen. Pero no replica la eliminación en el bucket de Outposts de destino. En otras palabras, no elimina la misma versión del objeto del bucket de Outposts de destino. Este comportamiento protege los datos de eliminaciones malintencionadas.

Cuando se añade una regla de replicación a un bucket de Outposts, la regla está activada de forma predeterminada, por lo que comienza a funcionar en cuanto se guarda.

En este ejemplo, se configura la replicación de los buckets de Outposts de origen y destino que están en Outposts distintos y son propiedad de la misma Cuenta de AWS. Se proporcionan ejemplos de cómo utilizar la consola de Amazon S3, la AWS Command Line Interface (AWS CLI), y AWS SDK for Java y AWS SDK for .NET. Para obtener más información sobre los permisos de Replicación de

S3 en Outposts entre cuentas, consulte Concesión de permisos cuando los buckets de Outposts de origen y destino pertenecen a diferentes Cuentas de AWS.

Para conocer los requisitos previos para configurar las reglas de replicación de S3 en Outposts, consulte Requisitos previos para crear reglas de replicación.

Uso de la consola de S3

Siga estos pasos para configurar una regla de replicación cuando el bucket de Amazon S3 en Outposts de destino esté en un Outposts distinto del bucket de Outposts de origen.

Si el bucket de Outposts de destino está en una cuenta distinta a la del bucket de Outposts de origen, se debe añadir una política de buckets al bucket de Outposts de destino para conceder al propietario de la cuenta del bucket de Outposts de origen permiso para replicar objetos en el bucket de Outposts de destino. Para obtener más información, consulte Concesión de permisos cuando los buckets de origen y destino son propiedad de diferentes Cuentas de AWS.

Para crear una regla de replicación

1. Inicie sesión en la AWS Management Console y abra la consola de Amazon S3 en https://console.aws.amazon.com/s3/.
2. En la lista Buckets de Outposts, seleccione el nombre del bucket para el que desea usar el bucket de origen.
3. Elija Administración, desplácese hacia abajo hasta Reglas de replicación y, a continuación, elija Crear regla de replicación.
4. En Nombre de la regla de replicación, escriba un nombre para la regla, de modo que pueda identificarla fácilmente más tarde. El nombre es obligatorio y debe ser único dentro del bucket.
5. En Estado, Habilitada está seleccionado de forma predeterminada. Una regla activada comienza a funcionar tan pronto se guarda. Si desea habilitar la regla más adelante, seleccione Deshabilitada.
6. En Prioridad, el valor de prioridad de la regla determina qué regla aplicar si hay reglas superpuestas. Cuando los objetos se incluyen en el ámbito de más de una de regla de replicación, S3 en Outposts utiliza estos valores de prioridad para evitar conflictos. De forma predeterminada, las reglas nuevas se agregan a la configuración de replicación con la prioridad más alta. Cuanto mayor sea el número, mayor será la prioridad.

   Para cambiar la prioridad de la regla, después de guardarla, elija el nombre de la regla en la lista de reglas de replicación, elija Acciones y, a continuación, elija Editar prioridad.

7. En Bucket de origen, tiene las siguientes opciones para establecer el origen de la replicación:

   - Para replicar todo el bucket, elija Aplicar a todos los objetos del bucket.
   - Para aplicar el filtrado de prefijos o etiquetas al origen de la replicación, elija Limitar el alcance de esta regla mediante uno o más filtros. Puede hacer uso combinado de un prefijo y etiquetas.
     - Para replicar todos los objetos que tengan el mismo prefijo, en Prefijo, introduzca un prefijo en el cuadro. Para limitar la replicación a todos los objetos que tienen nombres que empiezan con la misma cadena (por ejemplo,), use el filtro Prefijo`pictures`.

       Si escribe un prefijo que es el nombre de una carpeta, debe usar una / (barra inclinada) como último carácter (por ejemplo, `pictures/`).
     - Para replicar todos los objetos que tienen una o varias etiquetas de objeto, elija Agregar etiqueta y escriba el par clave-valor en los cuadros. Para agregar otra etiqueta, repita el procedimiento. Para obtener más información acerca de las etiquetas de objeto, consulte Agregar etiquetas para los buckets de S3 en Outposts.
8. Para acceder a su bucket de origen de S3 en Outposts para replicarlo, en Nombre del punto de acceso de origen, elija un punto de acceso que esté adjunto al bucket de origen.
9. En Destino, elija el ARN del punto de acceso del bucket de Outposts de destino donde desea que S3 en Outposts replique objetos. Los buckets de Outposts de destino pueden estar en diferentes Cuenta de AWS o dentro de la misma región que el bucket de Outposts de origen.

   Si el bucket de destino está en una cuenta distinta a la del bucket de Outposts de origen, se debe añadir una política de buckets al bucket de Outposts de destino para conceder al propietario de la cuenta del bucket de Outposts de origen permiso para replicar objetos en el bucket de Outposts de destino. Para obtener más información, consulte Concesión de permisos cuando los buckets de Outposts de origen y destino pertenecen a diferentes Cuentas de AWS.

   > **Note**
   >
   > Si el control de versiones no está habilitado en el bucket de Outposts de destino, recibirá una advertencia con el botón Habilitar el control de versiones. Elija este botón para activar el control de versiones en el bucket.
10. Configure un rol de servicio de AWS Identity and Access Management (IAM) que pueda asumir S3 en Outposts para reproducir objetos en su nombre.

Para configurar un rol de IAM, en Rol de IAM, lleve a cabo una de las siguientes acciones:

- Para que S3 en Outposts cree un nuevo rol de IAM para la configuración de replicación, Elegir entre los roles de IAM existentes y, a continuación, elija Crear un nuevo rol. Cuando se guarda la regla, se genera una política nueva para el rol de IAM que coincide con los buckets de Outposts de origen y destino elegidos. Le recomendamos que elija Crear un nuevo rol.
- También puede elegir usar un rol de IAM existente. Si lo hace, debe elegir un rol que conceda a S3 en Outposts los permisos necesarios para la replicación. La replicación dará un error si este rol no concede a S3 en Outposts permisos suficientes para seguir la regla de replicación.

  Para escoger un rol existente, marque Elegir entre los roles de IAM existentes y, a continuación, seleccione el nombre en el menú desplegable. También puede elegir Ingresar un ARN de rol de IAM y, a continuación, escribir el nombre de recurso de Amazon (ARN) del rol de IAM.

> **Important**
>
> Cuando añada una regla de replicación a un bucket de S3 en Outposts, debe tener los permisos `iam:CreateRole` y `iam:PassRole` para poder crear y pasar el rol de IAM que concede los permisos de replicación de S3 en Outposts. Para obtener más información, consulte Concesión de permisos a un usuario para transferir un rol a un servicio de Servicio de AWS en la Guía del usuario de IAM.

11. Todos los objetos de los buckets de Outposts están cifrados de forma predeterminada. Para obtener más información sobre el cifrado de S3 en Outposts, consulte Cifrado de datos en S3 en Outposts. Solo se pueden replicar los objetos cifrados en el servidor con claves administradas por Amazon S3 (SSE-S3). No se admite la replicación de objetos cifrados en el servidor con claves de AWS Key Management Service (AWS KMS) (SSE-KMS) o cifrado del lado del servidor con claves de cifrado proporcionadas por el cliente (SSE-C).
12. Habilite las siguientes opciones adicionales al configurar la regla de replicación, según sea necesario:

    - Si desea habilitar métricas de replicación de S3 en Outposts en la configuración de replicación, seleccione Métricas de replicación. Para obtener más información, consulte Monitoreo del progreso con métricas de replicación.

- Si desea habilitar la replicación de marcador de eliminación en la configuración de replicación, seleccione Delete marker replication (Eliminar replicación de marcadores). Para obtener más información, consulte Cómo afectan las operaciones de eliminación a la replicación.
- Si desea replicar los cambios de metadatos realizados en las réplicas en los objetos de origen, seleccione Sincronización de modificación de réplicas. Para obtener más información, consulte Estado de replicación si la sincronización de modificación de réplica de Amazon S3 en Outposts está habilitada.

13. Para terminar, seleccione Crear regla.

Después de guardar la regla, puede editarla, habilitarla, deshabilitarla o eliminarla. Para ello, vaya a la pestaña Administración del bucket de Outposts de origen, desplácese hacia abajo hasta la sección Reglas de replicación, elija su regla y, a continuación, Editar regla.

Utilización de la AWS CLI

Para utilizar la AWS CLI con el objetivo de configurar la replicación cuando los buckets de Outposts de origen y destino son propiedad de la misma Cuenta de AWS, debe hacer lo siguiente:

- Crear buckets de Outposts de origen y destino.
- Habilitar el control de versiones en ambos buckets.
- Crear un rol de IAM que conceda permisos a S3 en Outposts para replicar objetos.
- Agregar la configuración de replicación al bucket de Outposts de origen.

Para verificar la configuración, debe probarla.

Para configurar la replicación cuando los buckets de Outposts de origen y destino son propiedad de la misma Cuenta de AWS

1. Configure un perfil de credenciales para la AWS CLI. En este ejemplo, usamos el nombre de perfil `acctA`. Para obtener información acerca de la configuración de perfiles de credenciales, consulte Perfiles con nombre en la Guía del usuario de la AWS Command Line Interface.

   > **Important**
   >
   > Los perfiles utilizados para este ejercicio tienen que tener los permisos necesarios. Por ejemplo, en la configuración de replicación debe especificar el rol de servicio de IAM que puede asumir S3 en Outposts. Solo puede hacer esto si el perfil que utiliza tiene los

permisos `iam:CreateRole` y `iam:PassRole`. Para obtener más información, consulte Concesión de permisos a un usuario para transferir un rol a un servicio de Servicio de AWS en la Guía del usuario de IAM. Si utiliza credenciales de administrador para crear un perfil con nombre, el perfil con nombre tendrá el permiso necesario para realizar todas las tareas.

2. Cree un bucket de *origen* y habilite el control de versiones. El siguiente comando `create-bucket` crea un bucket *SOURCE-OUTPOSTS-BUCKET* en la región Este de EE. UU. (Norte de Virginia) (`us-east-1`). Para usar este comando, sustituya *user input placeholders* por su información.

```
aws s3control create-bucket --bucket SOURCE-OUTPOSTS-BUCKET --outpost-id SOURCE-OUTPOST-ID --profile acctA --region us-east-1
```

El siguiente comando `put-bucket-versioning` habilita el control de versiones en el bucket *SOURCE-OUTPOSTS-BUCKET*. Para usar este comando, sustituya *user input placeholders* por su información.

```
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. Cree un bucket de *destino* y habilite el control de versiones. El siguiente comando `create-bucket` crea un bucket *DESTINATION-OUTPOSTS-BUCKET* en la región Oeste de EE. UU. (Oregón) (`us-west-2`). Para usar este comando, sustituya *user input placeholders* por su información.

> **Note**
>
> Para establecer la configuración de replicación cuando los buckets de Outposts de origen y destino están en la misma Cuenta de AWS, debe utilizar el mismo perfil. En este ejemplo se utiliza `acctA`. Para probar la configuración de replicación cuando los buckets son propiedad de diferentes Cuentas de AWS, debe especificar diferentes perfiles para cada bucket.

```
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
```

El siguiente comando `put-bucket-versioning` habilita el control de versiones en el bucket *DESTINATION-OUTPOSTS-BUCKET*. Para usar este comando, sustituya *user input placeholders* por su información.

```
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. Cree un rol de servicio de IAM. Más adelante en la configuración de la replicación, agregue este rol de servicio al bucket *SOURCE-OUTPOSTS-BUCKET*. S3 en Outposts asume este rol para replicar objetos en su nombre. Crea el rol de IAM en dos pasos:

   a. Cree un rol de IAM.

      i. Copie la siguiente política de confianza y guárdela en un archivo llamado `s3-on-outposts-role-trust-policy.json` en el directorio actual en su equipo local. Esta política concede permisos a la entidad principal de servicio de S3 en Outposts para asumir el rol de servicio.

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

      ii. Ejecute el siguiente comando de para crear el rol. Reemplace los *user input placeholders* con su propia información.

```
aws iam create-role --role-name replicationRole --assume-role-policy-
document file://s3-on-outposts-role-trust-policy.json --profile acctA
```

b. Asocie una política de permisos al rol de servicio.

i. Copie la siguiente política de permisos y guárdela en un archivo llamado `s3-on-outposts-role-permissions-policy.json` en el directorio actual en su equipo local. Esta política concede permisos para varias acciones de buckets y objetos de S3 en Outposts. Para utilizar esta política, sustituya *`user input placeholders`* por su información.

```
{
   "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. Ejecute el siguiente comando para crear una política y asociarla al rol. Reemplace los *user input placeholders* con su propia información.

```
aws iam put-role-policy --role-name replicationRole --policy-
document file://s3-on-outposts-role-permissions-policy.json --policy-
name replicationRolePolicy --profile acctA
```

5. Agregue una configuración de replicación al bucket *SOURCE-OUTPOSTS-BUCKET*.

a. Si bien la API de S3 en Outposts requiere la configuración de replicación en formato XML, la AWS CLI requiere que especifique la configuración de reproducción en formato JSON. Guarde la siguiente JSON en un archivo denominado `replication.json` en el directorio local en su equipo. Para usar esta configuración, sustituya *user input placeholders* por su información.

```
{
  "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. Ejecute el siguiente comando `put-bucket-replication` para añadir la configuración de replicación al bucket de Outposts de origen. Para usar este comando, sustituya *user input placeholders* por su información.

```
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. Para recuperar la configuración de replicación, utilice el comando `get-bucket-replication`. Para usar este comando, sustituya *`user input placeholders`* por su información.

```
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. Pruebe la configuración en la consola de Amazon S3:

a. Inicie sesión en la AWS Management Console y abra la consola de Amazon S3 en https://console.aws.amazon.com/s3/.

b. En el bucket *`SOURCE-OUTPOSTS-BUCKET`*, cree una carpeta llamada `Tax`.

c. Agregue objetos de ejemplo a la carpeta `Tax` en el bucket *`SOURCE-OUTPOSTS-BUCKET`*.

d. En el bucket *`DESTINATION-OUTPOSTS-BUCKET`*, compruebe lo siguiente:

- S3 en Outposts ha replicado los objetos.

> **Note**
>
> El tiempo que S3 en Outposts tarda en replicar un objeto depende del tamaño del objeto. Para obtener más información sobre cómo ver el estado de la replicación, consulte Obtención de información del estado de replicación.

- En la pestaña Propiedades, el Estado de replicación se fija en Replica (lo que lo identifica como un objeto de réplica).

Administración de la replicación

En esta sección se describen opciones de configuración de replicación adicionales que están disponibles en S3 en Outposts, cómo determinar el estado de replicación y cómo solucionar problemas de replicación. Para obtener información acerca de la configuración de replicación principal, consulte Configuración de la replicación.

Temas

- Monitoreo del progreso con métricas de replicación

- Obtención de información del estado de replicación
- Solución de problemas de replicación
- Uso de EventBridge para la replicación de S3 en Outposts

Monitoreo del progreso con métricas de replicación

Replicación de S3 en Outposts proporciona métricas detalladas para las reglas de replicación en la configuración de la replicación. Las métricas de replicación permiten monitorear el progreso de la replicación en intervalos de 5 minutos realizando el seguimiento de los bytes pendientes de replicación, la latencia de replicación y las operaciones pendientes de replicación. También puede configurar Amazon EventBridge para recibir notificaciones de errores de replicación para ayudarlo a solucionar los problemas de configuración.

Si las métricas de replicación están habilitadas, Replicación de S3 en Outposts publica las siguientes métricas en Amazon CloudWatch:

- Bytes pendientes de replicación: número total de bytes de objetos pendientes de replicación para una regla de replicación determinada.
- Latencia de replicación: número máximo de segundos durante los cuales los buckets de destino de replicación están detrás del bucket de origen para una regla de replicación determinada.
- Operaciones pendientes de replicación: número de operaciones pendientes de replicación para una regla de replicación determinada. Las operaciones incluyen objetos, marcadores de eliminación y etiquetas.

> **Note**
>
> Las métricas de Replicación de S3 en Outposts se facturan al mismo precio que las métricas personalizadas de CloudWatch. Para obtener más información, consulte los precios de CloudWatch.

Obtención de información del estado de replicación

El estado de replicación puede ayudar a determinar el estado actual de un objeto que replica Amazon S3 en Outposts. El estado de replicación de un objeto de origen devolverá `PENDING`, `COMPLETED`, o `FAILED`. Se devolverá el estado de replicación de una réplica `REPLICA`.

Información general sobre el estado de replicación

En un supuesto de replicación, tiene un bucket de origen en el que se configura la replicación y un bucket de destino donde S3 en Outposts replica los objetos. Cuando solicita un objeto (utilizando `GetObject`) o los metadatos de un objeto (utilizando `HeadObject`) de estos buckets, S3 en Outposts devuelve el encabezado `x-amz-replication-status` en la respuesta, del siguiente modo:

- Si solicitar un objeto del bucket de origen, S3 en Outposts devuelve el encabezado `x-amz-replication-status` si el objeto de su solicitud cumple los requisitos para la replicación.

  Por ejemplo, supongamos que especifica el prefijo del objeto `TaxDocs` en la configuración de replicación para indicar a S3 en Outposts que replique solo objetos con el prefijo de nombre de clave `TaxDocs`. Cualquier objeto que cargue que tenga este prefijo de nombre de clave, por ejemplo, `TaxDocs/document1.pdf` se replicará. Para solicitudes de objetos con este prefijo de nombre de clave, S3 en Outposts devuelve el encabezado `x-amz-replication-status` con uno de los siguientes valores para el estado de replicación del objeto: `PENDING`, `COMPLETED` o `FAILED`.

  > **Note**
  >
  > Si la replicación de objetos genera un error después de cargar un objeto, no puede volver a intentar la replicación. Deberá cargar de nuevo el objeto. Los objetos pasan a un estado `FAILED` para problemas como la falta de permisos del rol de replicación o de permisos del bucket. En el caso de errores temporales, como cuando un bucket o su Outpost no están disponibles, el estado de replicación no pasa a `FAILED`, sino que permanece como `PENDING`. Después de que el recurso vuelva a estar en línea, S3 en Outposts reanuda la replicación de esos objetos.

- Cuando solicita un objeto desde un bucket de destino, si el objeto de la solicitud es una réplica creada por S3 en Outposts, S3 on Outposts devuelve el encabezado `x-amz-replication-status` con el valor `REPLICA`.

> **Note**
>
> Antes de eliminar un objeto del bucket de origen que tiene activada la replicación, revise el estado de replicación del objeto para asegurarse de que el objeto haya sido replicado.

Estado de replicación si la sincronización de modificación de réplica de Amazon S3 en Outposts está habilitada

Cuando las reglas de replicación habilitan la sincronización de la modificación de réplicas de S3 en Outposts, las réplicas pueden informar estados distintos de `REPLICA`. Si los cambios de metadatos están en proceso de replicación, el encabezado `x-amz-replication-status` para la réplica devuelve `PENDING`. Si la sincronización de modificación de réplica no replica metadatos, el encabezado para la réplica devuelve `FAILED`. Si los metadatos se replican correctamente, el encabezado para la réplica devuelve el valor `REPLICA`.

Solución de problemas de replicación

Si las réplicas de objetos no aparecen en el bucket de Amazon S3 en Outposts de destino después de configurar la replicación, use estos consejos de solución de problemas para identificar y solucionar los problemas.

- El tiempo que tarda S3 en Outpost en replicar un objeto depende de diferentes factores, como la distancia entre los Outposts de origen y destino, y el tamaño del objeto.

  También puede comprobar el estado de replicación del objeto de origen. Si el estado de replicación del objeto es `PENDING`, significa que S3 en Outposts no ha completado la replicación. Si el estado de replicación del objeto es `FAILED`, compruebe la configuración de replicación establecida en el bucket de origen.
- En la configuración de replicación en el bucket de origen, verifique lo siguiente:
  - El nombre de recurso de Amazon (ARN) del punto de acceso del bucket de destino es correcto.
  - El prefijo de nombre de clave sea correcto. Por ejemplo, si establece la configuración para replicar objetos con el prefijo `Tax`, entonces, solo se replicarán los objetos con nombres de clave como `Tax/document1` o `Tax/document2`. No se replicará un objeto con el nombre de clave `document3`.
  - El estado es `Enabled`.
- Compruebe que el control de versiones no se ha suspendido en ningún bucket. Ambos buckets de origen y destino deben tener habilitado el control de versiones.
- Si el bucket de destino pertenece a otra Cuenta de AWS, compruebe que el propietario del bucket tenga una política de bucket en el bucket de destino que permita al propietario del bucket de origen replicar objetos. Para ver un ejemplo, consulte Concesión de permisos cuando los buckets de Outposts de origen y destino pertenecen a diferentes Cuentas de AWS.

- Si la réplica de un objeto no aparece en el bucket de destino, los siguientes problemas podría haber impedido la replicación:
  - S3 en Outposts no replica un objetos de un bucket de origen si es una réplica creada por otra configuración de replicación. Por ejemplo, si establece una configuración de replicación del bucket A en el bucket B y, luego, en el bucket C, S3 en Outposts no replica las réplicas de objetos del bucket B en el bucket C.

    Si desea replicar objetos del bucket A en el bucket B y en el bucket C, defina varios destinos de bucket en diferentes reglas de replicación para la configuración de replicación del bucket de origen. Por ejemplo, cree dos reglas de replicación en el bucket de origen A, con una regla para replicar en el bucket de destino B y la otra regla para replicar en el bucket de destino C.
  - Un propietario del bucket de origen puede conceder permisos a otras Cuentas de AWS para cargar objetos. De forma predeterminada, el propietario del bucket de origen no tiene permisos sobre los objetos creados por otras cuentas. La configuración de replicación solo replica los objetos para los que el propietario del bucket de origen tiene permisos de acceso. Para evitar problemas de replicación, el propietario del bucket de origen puede conceder permisos a otras Cuentas de AWS para crear objetos con la condición de que tengan permisos de acceso explícitos para esos objetos. Para ver una política de ejemplo, consulte Conceder permisos entre cuentas para cargar objetos al mismo tiempo que se garantiza que el propietario del bucket tenga el control total.
- Supongamos que en la configuración de replicación añade una regla para replicar un subconjunto de objetos con una etiqueta específica. En este caso, debe asignar la clave de etiqueta y el valor específicos en el momento de crear el objeto para que S3 en Outposts replique el objeto. Si primero crea un objeto y luego agrega la etiqueta en el objeto existente, S3 en Outposts no replica el objeto.
- La replicación devuelve un error si la política de bucket deniega el acceso a la función de replicación para cualquiera de las siguientes acciones:

  Bucket de origen:

```
"s3-outposts:GetObjectVersionForReplication",
"s3-outposts:GetObjectVersionTagging"
```

  Buckets de destino:

```
"s3-outposts:ReplicateObject",
"s3-outposts:ReplicateDelete",
```

```
"s3-outposts:ReplicateTags"
```

- Amazon EventBridge pueden enviarle notificaciones cuando los objetos no se repliquen en su Outposts de destino. Para obtener más información, consulte Uso de EventBridge para la replicación de S3 en Outposts.

Uso de EventBridge para la replicación de S3 en Outposts

Amazon S3 en Outposts se integra con Amazon EventBridge y utiliza el espacio de nombres `s3-outposts`. EventBridge es un servicio de bus de eventos sin servidor que se puede utilizar para conectar las aplicaciones con datos de varios orígenes. Para obtener más información, consulte What is Amazon EventBridge? (¿Qué es Amazon EventBridge?) en la Guía del usuario de Amazon EventBridge.

Puede configurar Amazon EventBridge para recibir notificaciones de eventos de error de replicación para ayudar a solucionar cualquier problema de configuración de la replicación. EventBridge pueden notificarle en instancias cuando los objetos no se repliquen en su Outposts de destino. Para obtener más información sobre el estado actual de un objeto que se replica, consulte Información general sobre el estado de replicación.

S3 en Outposts puede enviar eventos a EventBridge cada vez que se produzcan determinados eventos en el bucket. A diferencia de otros destinos, no es necesario seleccionar qué tipos de eventos desea entregar. Puede utilizar las reglas de EventBridge para dirigir los eventos hacia destinos adicionales. Una vez habilitado EventBridge, S3 en Outposts envía todos los eventos que sigan a EventBridge.

Tipo de evento	Descripción	Espacio de nombres
`Operation FailedReplication`	No se ha podido replicar un objeto dentro de una regla de replicación. Para obtener más información sobre los errores de Replicación de S3 en Outposts, consulte Uso de EventBridge para ver los motivos de error de Replicación de S3 en Outposts.	`s3-outposts`

Uso de EventBridge para ver los motivos de error de Replicación de S3 en Outposts

En la siguiente tabla se muestran los motivos de error de Replicación de S3 en Outposts. Puede configurar una regla de EventBridge para publicar y ver el motivo de error a través de Amazon Simple Queue Service (Amazon SQS), Amazon Simple Notification Service (Amazon SNS), AWS Lambda o Registros de Amazon CloudWatch. Para obtener más información sobre los permisos necesarios para utilizar estos recursos de EventBridge, consulte el tema sobre el uso de las políticas basadas en recursos para EventBridge.

Motivo de error de replicación	Descripción
`AssumeRoleNotPermitted`	S3 en Outposts no puede asumir el rol (de IAM) AWS Identity and Access Managemen t que se especifica en la configuración de replicación.
`DstBucketNotFound`	S3 en Outposts no encuenta el bucket de destino especificado en la configuración de replicación.
`DstBucketUnversioned`	El control de versiones no está habilitado en el bucket de destino de Outposts. Habilite el control de versiones en el bucket de destino para replicar objetos con Replicación de S3 en Outposts.
`DstDelObjNotPermitted`	S3 en Outposts no puede replicar lo que se ha eliminado en el bucket de destino. Es posible que falte el permiso `s3-outpos ts:ReplicateDelete` para el bucket de destino.
`DstMultipartCompleteNotPermitted`	S3 en Outposts no puede completar la carga multiparte de objetos en el bucket de destino. Es posible que falte el permiso `s3-outpos ts:ReplicateObject` para el bucket de destino.

Motivo de error de replicación	Descripción
`DstMultipartInitNotPermitted`	S3 en Outposts no puede iniciar una carga multiparte de objetos en el bucket de destino. Es posible que falte el permiso `s3-outposts:ReplicateObject` para el bucket de destino.
`DstMultipartPartUploadNotPermitted`	S3 en Outposts no puede iniciar una carga multiparte de objetos en el bucket de destino. Es posible que falte el permiso `s3-outposts:ReplicateObject` para el bucket de destino.
`DstOutOfCapacity`	S3 en Outposts no puede replicarse en el Outpost de destino porque el Outpost no tiene capacidad de almacenamiento de S3.
`DstPutObjNotPermitted`	S3 en Outposts no puede replicar objetos en el bucket de destino. Es posible que falte el permiso `s3-outposts:ReplicateObject` para el bucket de destino.
`DstPutTaggingNotPermitted`	S3 en Outposts no puede replicar etiquetas de objetos en el bucket de destino. Es posible que falte el permiso `s3-outposts:ReplicateObject` para el bucket de destino.
`DstVersionNotFound`	S3 en Outposts no encuentra la versión del objeto requerida en el bucket de destino para replicar los metadatos de esa versión del objeto.
`SrcBucketReplicationConfigMissing`	S3 en Outposts no encuentra una configuración de replicación para el punto de acceso asociado al bucket de Outposts de origen.

Motivo de error de replicación	Descripción
`SrcGetObjNotPermitted`	S3 en Outposts no puede acceder al objeto del bucket de origen para replicarlo. Es posible que falte el permiso `s3-outposts:GetObjectVersionForReplication` para el bucket de origen.
`SrcGetTaggingNotPermitted`	S3 en Outposts no puede acceder a la etiqueta de objeto del bucket de origen. Es posible que falte el permiso `s3-outposts:GetObjectVersionTagging` para el bucket de origen.
`SrcHeadObjectNotPermitted`	S3 en Outposts no puede recuperar los metadatos de objetos del bucket de origen. Es posible que falte el permiso `s3-outposts:GetObjectVersionForReplication` para el bucket de origen.
`SrcObjectNotEligible`	El objeto no es apto para la replicación. Los objetos y sus etiquetas de objetos no coinciden con la configuración de replicación.

Para obtener más información acerca de la resolución de problemas de replicación, consulte los siguientes temas:

- Creación de un rol de IAM
- Solución de problemas de replicación

Monitoreo de EventBridge con CloudWatch

Para el monitoreo, se ha integrado Amazon CloudWatch con Amazon EventBridge. EventBridge envía automáticamente métricas a CloudWatch cada minuto. Estas métricas incluyen el número de eventos que coincide con una regla y el número de veces que una regla invoca un objetivo. Cuando se ejecuta una regla en EventBridge, se invocan todos los destinos asociados a la regla. Puede monitorear su comportamiento en EventBridge a través de CloudWatch de las siguientes maneras.

- Puede monitorear las métricas de EventBridge disponibles para sus reglas de EventBridge desde el panel de CloudWatch. A continuación, puede utilizar las funciones de CloudWatch, como las alarmas de CloudWatch, para configurar alarmas en determinadas métricas. Si esas métricas alcanzan los valores límite personalizados que ha especificado en las alarmas, recibirá notificaciones y podrá tomar las medidas pertinentes.
- Puede configurar Registros de Amazon CloudWatch como destino de su regla de EventBridge. A continuación, EventBridge crea flujos de registro y CloudWatch Logs almacena el texto de los eventos como entradas de registro. Para obtener más información, consulte EventBridge y CloudWatch Logs.

Para obtener más información sobre la depuración de eventos de archivo y la entrega de evenros de EventBridge, consulte los siguientes temas:

- Política de reintentos de eventos y uso de colas de mensajes fallidos
- Archivado de eventos de EventBridge

Uso compartido de S3 en Outposts con AWS RAM

Amazon S3 en Outposts admite el uso compartido de la capacidad de S3 en varias cuentas de una organización mediante AWS Resource Access Manager (AWS RAM). Con el uso compartido de S3 en Outposts, puede permitir que otros creen y administren los buckets, los puntos de conexión y los puntos de acceso en su Outpost.

En este tema se muestra cómo utilizar AWS RAM para compartir S3 en Outposts y los recursos relacionados con otra Cuenta de AWS en su organización de AWS.

Requisitos previos

- La cuenta propietaria de Outpost tiene configurada una organización en AWS Organizations. Para obtener más información, consulte Creación de una organización en la Guía del usuario de AWS Organizations.
- La organización incluye la Cuenta de AWS con la que desea compartir la capacidad de S3 en Outposts. Para obtener más información, consulte Envío de invitaciones aCuentas de AWS en la Guía del usuario de AWS Organizations.
- Seleccione una de las siguientes opciones que desea compartir. Se debe seleccionar el segundo recurso (las Subnets [Subredes] u Outposts [Outposts]) para que también se pueda acceder a

los puntos de conexión. Los puntos de conexión son un requisito de red para acceder a los datos almacenados en S3 en Outposts.

Opción 1	Opción 2
S3 en Outposts Permite que el usuario cree buckets en sus Outposts y puntos de acceso y que agregue objetos en esos buckets.	S3 en Outposts Permite que el usuario cree buckets en sus Outposts y puntos de acceso y que agregue objetos en esos buckets.
Subredes Permite que el usuario utilice la nube privada virtual (VPC) y los puntos de conexión asociados a la subred.	Outposts Permite que el usuario vea los gráficos de capacidad de S3 y la página de inicio de la consola de AWS Outposts. También permite que los usuarios creen subredes en Outposts compartidos y que creen puntos de conexión.

Procedimiento

1. Inicie sesión en la AWS Management Console mediante la Cuenta de AWS propietaria del Outpost y, a continuación, abra la consola de AWS RAM en https://console.aws.amazon.com/ram.
2. Asegúrese de haber habilitado la opción para compartir AWS Organizations en AWS RAM. Para obtener información, consulte Habilitar el uso compartido con AWS Organizations en la Guía del usuario de AWS RAM.
3. Utilice la opción 1 o la opción 2 en requisitos previos para crear un recurso compartido. Si tiene varios recursos de S3 en Outposts, seleccione los nombres de recurso de Amazon (ARN) de los recursos que desea compartir. Para habilitar los puntos de conexión, comparta la subred u Outpost.

   Para obtener información sobre cómo crear un recurso compartido, consulte Creación de un recurso compartido en la Guía del usuario de AWS RAM.
4. La Cuenta de AWS con la que compartió sus recursos debería poder utilizar S3 en Outposts. Según la opción que seleccionó en los requisitos previos, proporcione la siguiente información al usuario de la cuenta:

Opción 1	Opción 2
El ID de Outpost	El ID de Outpost
El ID de la VPC	
El ID de subred	
El ID del grupo de seguridad	

> **Note**
>
> El usuario puede confirmar que los recursos se compartieron con ellos mediante la consola de AWS RAM, la AWS Command Line Interface (AWS CLI), los SDK de AWS o la API de REST. El usuario puede ver sus recursos compartidos existentes con el comando de la CLI get-resource-shares.

Ejemplos de uso

Una vez que haya compartido sus recursos de S3 en Outposts con otra cuenta, esa cuenta puede administrar los buckets y los objetos en su Outpost. Si compartió el recurso Subnets (Subredes), luego esa cuenta puede utilizar el punto de conexión que ha creado. En los siguientes ejemplos, se muestra cómo un usuario puede utilizar AWS CLI para interactuar con su Outpost después de compartir estos recursos.

Example : crear un bucket

En el siguiente ejemplo, se crea un bucket denominado *amzn-s3-demo-bucket1* en el Outpost *op-01ac5d28a6a232904*. Antes de utilizar este comando, reemplace cada *user input placeholder* con los valores adecuados para su caso de uso.

```
aws s3control create-bucket --bucket amzn-s3-demo-bucket1 --outpost-
id op-01ac5d28a6a232904
```

Para obtener más información acerca de este comando, consulte create-bucket en la Referencia de AWS CLI.

Example : crear un punto de acceso

En el siguiente ejemplo, se crea un punto de acceso en un Outpost mediante los parámetros de ejemplo de la siguiente tabla. Antes de utilizar este comando, reemplace estos valores *user input placeholder* y el código de la Región de AWS con los valores adecuados para su caso de uso.

Parámetro	Valor
ID de cuenta	*111122223333*
Nombre del punto de acceso	*example-outpost-access-point*
ID de Outpost	*op-01ac5d28a6a232904*
Nombre del bucket del Outpost	*amzn-s3-demo-bucket1*
ID de VPC	*vpc-1a2b3c4d5e6f7g8h9*

> **Note**
>
> El parámetro de ID de la cuenta debe ser el ID de la Cuenta de AWS del propietario del bucket, que es el usuario compartido.

```
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/amzn-s3-demo-bucket1 \
--vpc-configuration VpcId=vpc-1a2b3c4d5e6f7g8h9
```

Para obtener más información acerca de este comando, consulte create-access-point en la Referencia de AWS CLI.

Example : cargar un objeto

En el siguiente ejemplo, se carga el archivo *my_image.jpg* desde el sistema de archivos local del usuario a un objeto denominado *images/my_image.jpg* a través del punto de acceso *example-outpost-access-point* en el Outpost *op-01ac5d28a6a232904*, propiedad de la cuenta de

AWS *111122223333*. Antes de utilizar este comando, reemplace estos valores *user input placeholder* y el código de la Región de AWS con los valores adecuados para su caso de uso.

```
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
```

Para obtener más información acerca de este comando, consulte put-object en la Referencia de AWS CLI.

> **Note**
>
> Si esta operación devuelve el error Resource not found (Recurso no encontrado) o no responde, es posible que la VPC no tenga un punto de conexión compartido.
> Para comprobar si hay un punto de conexión compartido, utilice el comando de la AWS CLI list-shared-endpoints (puntos finales de lista compartida). Si no hay un punto de conexión compartido, trabaje con el propietario del Outpost para crear uno. Para obtener más información, consulte ListSharedEndpoints en la Referencia de la API de Amazon Simple Storage Service.

Example : crear un punto de conexión

En el siguiente ejemplo, se crea un punto de conexión en un Outpost compartido. Antes de utilizar este comando, sustituya los valores *user input placeholder* para el ID de Outpost, el ID de subred y el ID del grupo de seguridad con los valores adecuados para su caso de uso.

> **Note**
>
> El usuario puede realizar esta operación solo si el recurso compartido incluye el recurso de Outposts.

```
aws s3outposts create-endpoint --outposts-id op-01ac5d28a6a232904 --subnet-id XXXXXX --
security-group-id XXXXXXX
```

Para obtener más información acerca de este comando, consulte create-endpoint en la Referencia de AWS CLI.

Otros Servicios de AWS que usan S3 en Outposts

Otros Servicios de AWS que se ejecutan de forma local en su AWS Outposts también pueden utilizar la capacidad de Amazon S3 en Outposts. En Amazon CloudWatch, el espacio de nombres `S3Outposts` muestra métricas detalladas de los buckets dentro de S3 en Outposts, pero estas métricas no incluyen el uso de otros Servicios de AWS. Para administrar la capacidad de S3 en Outposts que consumen otros Servicios de AWS, consulte la información de la tabla siguiente.

Servicio de AWS	Descripción	Más información
Simple Storage Service (Amazon S3)	Todo uso directo de S3 en Outposts tiene una métrica de CloudWatch de bucket y una cuenta coincidentes.	Consulte Métricas
Amazon Elastic Block Store (Amazon EBS)	Para Amazon EBS on Outposts, puede elegir un Outpost de AWS como destino de instantáneas y almacenarlo localmente en su S3 en Outpost.	Más información
Amazon Relational Database Service (Amazon RDS)	Puede utilizar las copias de seguridad locales de Amazon RDS para almacenar sus copias de seguridad de RDS localmente en su Outpost.	Más información

Monitoreo de S3 en Outposts

Con Amazon S3 en Outposts, puede crear buckets de S3 en Outposts de AWS y almacenar y recuperar fácilmente objetos en las instalaciones para las aplicaciones que requieren acceso local a los datos, procesamiento local de los datos y residencia de los datos. S3 en Outposts proporciona una nueva clase de almacenamiento, S3 Outposts (`OUTPOSTS`), que utiliza las API de Amazon S3 y está diseñada para almacenar datos de manera duradera y redundante en múltiples dispositivos y servidores de AWS Outposts. Usted se comunica con su bucket de Outpost mediante un punto de acceso y una conexión de punto de conexión a través de una nube privada virtual (VPC). Puede usar las mismas API y características en los buckets de Outposts que en buckets de Amazon S3, como políticas de acceso, cifrado y etiquetado. Puede utilizar S3 en Outposts a través de la AWS Management Console, AWS Command Line Interface (AWS CLI), AWS SDK o la API de REST. Para obtener más información, consulte ¿Qué es Amazon S3 en Outposts?

Para obtener más información sobre cómo administrar la capacidad de almacenamiento de Amazon S3 en Outposts, consulte los siguientes temas.

Temas

Administración de la capacidad de S3 en puestos avanzados con las métricas de Amazon CloudWatch

Para ayudar a administrar la capacidad fija de S3 en Outpost, recomendamos que cree alertas de CloudWatch que le digan cuándo la utilización del almacenamiento supera un umbral determinado. Para obtener más información acerca de las métricas de CloudWatch para S3 en Outposts, consulte Métricas de CloudWatch. Si no hay suficiente espacio para almacenar un objeto en su Outpost, la API devuelve una excepción de capacidad insuficiente (ICE). Para liberar espacio, puede crear alarmas de CloudWatch que desencadenen la eliminación explícita de datos o utilizar una política de vencimiento del ciclo de vida para hacer vencer los objetos. Para guardar datos antes de la eliminación, puede usar AWS DataSync para copiar datos desde el bucket de Amazon S3 en Outposts hasta un bucket de S3 en una Región de AWS. Para obtener más información acerca del uso de DataSync, consulte Introducción a AWS DataSync en la Guía del usuario de AWS DataSync.

Métricas de CloudWatch

El espacio de nombres `S3Outposts` incluye las siguientes métricas de buckets de Amazon S3 en Outposts. Puede monitorear el número total de bytes de S3 en Outposts aprovisionados, el total de bytes libres disponibles para los objetos y el tamaño total de todos los objetos para un bucket determinado. Existen métricas relacionadas con el bucket o la cuenta para todo el uso directo de S3. El uso indirecto de S3, como almacenar las instantáneas locales de Amazon Elastic Block Store o las copias de seguridad de Amazon Relational Database Service en un Outpost, consume capacidad de S3, pero no se incluye en las métricas relacionadas con el bucket o la cuenta. Para obtener más información acerca de las instantáneas locales de Amazon EBS, consulte Instantáneas locales de Amazon EBS en Outposts. Para ver el informe de costos de Amazon EBS, consulte https://console.aws.amazon.com/billing/.

> **Note**
>
> S3 en Outposts solo admite las siguientes métricas y ninguna otra métrica de Amazon S3. Dado que S3 en Outposts tiene un límite de capacidad fija, recomendamos crear alarmas de CloudWatch que le notifiquen cuando el uso del almacenamiento supere cierto umbral.

Métrica	Descripción	Periodo	Unidades	Tipo
`Outpost` `talByte`	La capacidad total aprovisionada en bytes para un Outpost.	5 minutos	Bytes	S3 on Outposts
`Outpostl` `eeBytes`	El recuento de bytes libres disponibles en Outposts para almacenar datos de clientes	5 minutos	Bytes	S3 on Outposts
`BucketU` `dBytes`	El tamaño total de todos los objetos para el bucket determinado.	5 minutos	Bytes	S3 en Outposts. Solo para uso directo de S3.
`Accountl` `edBytes`	El tamaño total de todos los objetos para la cuenta especificada de Outposts	5 minutos	Bytes	S3 en Outposts. Solo para uso directo de S3.
`BytesPe` `ingRepl` `ation`	Número total de bytes de objetos pendientes de replicación para una regla de replicación determinada. Para obtener más información sobre cómo activar las métricas de replicación, consulte el tema sobre cómo crear reglas de replicación entre Outposts.	5 minutos	Bytes	Opcional. Para S3 Replication en Outposts.
`Operati` `sPendin`	Número total de operaciones pendientes de replicación	5 minutos	Recuento	Opcional. Para S3 Replication en Outposts.

Métrica	Descripción	Periodo	Unidades	Tipo
`eplicat:` n	para una regla de replicación determinada. Para obtener más información sobre cómo activar las métricas de replicación, consulte el tema sobre cómo crear reglas de replicación entre Outposts.			
`Replica:` `onLaten`	Número actual de segundos de retraso durante los cuales el bucket de destino de replicación está detrás del bucket de origen para una regla de replicación determinada. Para obtener más información sobre cómo activar las métricas de replicación, consulte el tema sobre cómo crear reglas de replicación entre Outposts.	5 minutos	Segundos	Opcional. Para S3 Replication en Outposts.

Recepción de notificaciones de eventos de S3 en Outposts mediante Amazon CloudWatch Events

Puede utilizar CloudWatch Events para crear una regla para cualquier evento de API de Amazon S3 en Outposts. Al crear una regla, puede elegir recibir notificaciones a través de todos los destinos de CloudWatch compatibles, incluidos Amazon Simple Queue Service (Amazon SQS), Amazon Simple Notification Service (Amazon SNS) y AWS Lambda. Para obtener más información, consulte la lista de servicios de AWS que pueden ser destinos de CloudWatch Events en la Guía del usuario de Eventos de Amazon CloudWatch. Para elegir un servicio de destino para trabajar con S3 en Outposts, consulte Creación de una regla de CloudWatch Events que se desencadena en una llamada a la API de AWS con AWS CloudTrail en la Guía del usuario de Eventos de Amazon CloudWatch.

> **Note**
>
> Para las operaciones de objetos de S3 en Outposts, los eventos de llamada a la API de AWS enviados por CloudTrail solo coincidirán con sus reglas si tiene registros (opcionalmente con selectores de eventos) configurados para recibir dichos eventos. Para obtener más información, consulte Uso de archivos de registro de CloudTrail en la Guía del usuario de AWS CloudTrail.

Example

A continuación se muestra una regla de ejemplo para la operación de `DeleteObject`. Para utilizar esta regla de ejemplo, sustituya *amzn-s3-demo-bucket1* por el nombre del bucket de S3 en Outposts.

```
{
  "source": [
    "aws.s3-outposts"
  ],
  "detail-type": [
    "AWS API call through CloudTrail"
  ],
  "detail": {
    "eventSource": [
      "s3-outposts.amazonaws.com"
    ],
    "eventName": [
      "DeleteObject"
    ],
    "requestParameters": {
      "bucketName": [
        "amzn-s3-demo-bucket1"
      ]
    }
  }
}
```

Monitoreo de S3 en Outposts con registros de AWS CloudTrail

Amazon S3 en Outposts se integra con AWS CloudTrail, un servicio que proporciona un registro de las medidas adoptadas por un usuario, un rol o un servicio de Servicio de AWS en S3 en Outposts.

Puede utilizar AWS CloudTrail para obtener información sobre las solicitudes en el nivel de bucket y de objeto de S3 en Outpost para auditar y registrar su actividad de eventos de S3 en Outposts. Para activar los eventos de datos de CloudTrail para todos los buckets de Outposts o para una lista de buckets específicos de Outposts, debe crear un registro de seguimiento manualmente en CloudTrail. Para obtener más información sobre las entradas de archivos de registro de CloudTrail, consulte Entradas de archivo de registro de Amazon S3 en Outposts.

> **Note**
>
> - Una práctica recomendada consiste en crear una política de ciclo de vida para el bucket de Outposts de eventos de datos de AWS CloudTrail. Configure la política de ciclo de vida para eliminar periódicamente los archivos de registro tras el periodo de tiempo que necesite para auditarlos. Esto reduce la cantidad de datos que Amazon Athena analiza para cada consulta. Para obtener más información, consulte Configuración de un ciclo de vida en un bucket.
> - Para obtener ejemplos de cómo consultar los registros de CloudTrail, visite la publicación Analyze Security, Compliance, and Operational Activity Using AWS CloudTrail and Amazon Athena del Blog de big data de AWS.

Activar el registro de CloudTrail para objetos en un bucket de S3 en Outposts

Puede utilizar la consola de Amazon S3 para configurar una auditoría de AWS CloudTrail con el fin de registrar eventos de datos para objetos en un bucket de Amazon S3 en Outposts. CloudTrail permite que se registren operaciones de API en el nivel de objetos de S3 en Outsposts como, por ejemplo, `GetObject`, `DeleteObject` y `PutObject`. Estos eventos se denominan eventos de datos.

De forma predeterminada, los registros de seguimiento de CloudTrail no registran eventos de datos. Sin embargo, puede configurarlos para que registren eventos de datos en los buckets de S3 en Outposts que especifique o para registrar eventos de datos para todos los buckets de S3 en Outposts de su Cuenta de AWS. Para obtener más información, consulte Registro de llamadas a la API de Amazon S3 mediante AWS CloudTrail.

CloudTrail no rellena eventos de datos en el historial de eventos de CloudTrail. Además, no todas las operaciones de la API de nivel de bucket de S3 en Outposts se rellenan en el historial de eventos de CloudTrail. Para obtener más información acerca de cómo consultar los registros de CloudTrail,

consulte el artículo del Centro de conocimientos de AWS sobre el uso de patrones de filtro de registros de Amazon CloudWatch y Amazon Athena para consultar los registros de CloudTrail.

Para configurar un registro de seguimiento para que registre eventos de datos para un bucket de S3 en Outposts, puede utilizar la consola de AWS CloudTrail o la consola de Amazon S3. En caso de que esté configurando un registro de seguimiento con el fin de registrar eventos de datos para todos los buckets de S3 en Outposts en su Cuenta de AWS, es más fácil utilizar la consola de CloudTrail. Para obtener información sobre el uso de la consola de CloudTrail a fin de configurar un registro de seguimiento con el objetivo de registrar eventos de datos de S3 en Outposts, consulte Eventos de datos en la Guía del usuario de AWS CloudTrail.

> **Important**
>
> Se aplican cargos adicionales a los eventos de datos. Para obtener más información, consulte Precios de AWS CloudTrail.

En el siguiente procedimiento, se muestra cómo utilizar la consola de Amazon S3 a fin de configurar un registro de seguimiento de CloudTrail con el objetivo de registrar eventos de datos para un bucket de S3 en Outposts.

> **Note**
>
> La Cuenta de AWS que crea el bucket es su propietaria y la única que puede configurar eventos de datos de S3 en Outposts para enviarlos a AWS CloudTrail.

Para activar el registro de eventos de datos de CloudTrail para objetos en un bucket de S3 en Outposts

1. Inicie sesión AWS Management Console Management Console y abra la consola de Amazon S3 en https://console.aws.amazon.com/s3/.
2. En el panel de navegación izquierdo, elija Outposts buckets (Buckets de Outposts).
3. Elija el nombre el bucket de Outposts cuyos eventos de datos desea registrar mediante CloudTrail.
4. Seleccione Propiedades.
5. Vaya a la sección Eventos de datos de AWS CloudTrail y elija Configurar en CloudTrail.

Se abrirá la consola de AWS CloudTrail.

Puede crear un nuevo registro de seguimiento de CloudTrail o reutilizar uno existente y configurar eventos de datos de S3 en Outposts para que se registren en el seguimiento.

6. En la página Panel de la consola de CloudTrail, seleccione Crear un registro de seguimiento.
7. En la página Paso 1 Elegir atributos del registro de seguimiento, proporcione un nombre para el registro de seguimiento, elija un bucket de S3 para almacenar los registros de seguimiento, especifique cualquier otra configuración que desee y, a continuación, elija Siguiente.
8. En la página Paso 2 Elegir eventos de registro, en Tipo de evento, elija Eventos de datos.

   En Tipo de evento de datos, elija S3 Outposts. Elija Siguiente.

   > **Note**
   >
   > - Al crear un registro de seguimiento y configurar el registro de eventos de datos para S3 en Outposts, debe especificar el tipo de evento de datos correctamente.
   >   - Si usa la consola de CloudTrail, elija S3 Outposts como Tipo de evento de datos. Para obtener información acerca de cómo crear seguimientos en la consola de CloudTrail, consulte Creación y actualización de un seguimiento con la consola en la Guía del usuario de AWS CloudTrail. Si quiere obtener información sobre cómo configurar el registro de eventos de datos de S3 en Outposts en la consola de CloudTrail, consulte el punto Registrar eventos de datos para objetos de Amazon S3 en la guía del usuario de AWS CloudTrail.
   >   - Si usa la AWS Command Line Interface (AWS CLI) o los SDK de AWS, defina el campo `resources.type` como `AWS::S3Outposts::Object`. Para obtener más información sobre cómo registrar eventos de datos de S3 en Outposts con la AWS CLI, consulte Registrar eventos de S3 en Outposts en la Guía del usuario de AWS CloudTrail.
   > - Si utiliza la consola de CloudTrail o la consola de Amazon S3 para configurar un registro de seguimiento para registrar eventos de datos para un bucket de S3 en Outposts, la consola de Amazon S3 muestra que los registros de nivel de objeto están activados para el bucket.

9. En la página Paso 3 Revisar y crear, revise los atributos del registro de seguimiento y los eventos de registro que ha configurado. Después, seleccione Crear un registro de seguimiento.

Para desactivar el registro de eventos de datos de CloudTrail para objetos en un bucket de S3 en Outposts

1. Inicie sesión en la AWS Management Console y abra la consola de CloudTrail en https://console.aws.amazon.com/cloudtrail/.
2. En el panel de navegación situado a la izquierda, elija Registros de seguimiento.
3. Elija el nombre del registro de seguimiento que creo para registrar los eventos de su bucket de S3 en Outposts.
4. En la página de detalles del registro de seguimiento, seleccione Detener registro en la esquina superior derecha.
5. En el cuadro de diálogo que aparece, elija Detener registro.

Desarrollo con Amazon S3 en Outposts

Con Amazon S3 en Outposts, puede crear buckets de S3 en Outposts de AWS y almacenar y recuperar fácilmente objetos en las instalaciones para las aplicaciones que requieren acceso local a los datos, procesamiento local de los datos y residencia de los datos. S3 en Outposts proporciona una nueva clase de almacenamiento, S3 Outposts (OUTPOSTS), que utiliza las API de Amazon S3 y está diseñada para almacenar datos de manera duradera y redundante en múltiples dispositivos y servidores de AWS Outposts. Usted se comunica con su bucket de Outpost mediante un punto de acceso y una conexión de punto de conexión a través de una nube privada virtual (VPC). Puede usar las mismas API y características en los buckets de Outposts que en buckets de Amazon S3, como políticas de acceso, cifrado y etiquetado. Puede utilizar S3 en Outposts a través de la AWS Management Console, AWS Command Line Interface (AWS CLI), AWS SDK o la API de REST. Para obtener más información, consulte ¿Qué es Amazon S3 en Outposts?

Los siguientes temas proporcionan información acerca de cómo desarrollar con S3 en Outposts.

Temas

- Operaciones de la API de Amazon S3 en Outposts
- Configure el cliente de control de S3 para S3 en Outposts con SDK para Java
- Realización de solicitudes a S3 en Outposts mediante IPv6

Operaciones de la API de Amazon S3 en Outposts

En este tema, se enumeran las operaciones de la API de Amazon S3, Amazon S3 Control y Amazon S3 en Outposts que puede usar con Amazon S3 en Outposts

Temas

Operaciones de la API de Amazon S3 para administrar objetos

S3 en Outposts está diseñado para utilizar las mismas operaciones de la API de objetos que Amazon S3. Debe utilizar puntos de acceso para acceder a cualquier objeto de un bucket de Outpost. Al utilizar una operación de API de objetos con S3 en Outposts, proporciona el Nombre de recurso de Amazon (ARN) del punto de acceso de Outposts o el alias del punto de acceso. Para obtener más información acerca de los alias de punto de acceso, consulte Uso de un alias de estilo de bucket para su punto de acceso de bucket de S3 en Outposts.

Amazon S3 en Outposts admite las siguientes operaciones de la API de Amazon S3:

- AbortMultipartUpload
- CompleteMultipartUpload
- CopyObject
- CreateMultipartUpload
- DeleteObject
- DeleteObjects
- DeleteObjectTagging
- GetObject
- GetObjectTagging
- HeadBucket
- HeadObject
- ListMultipartUploads
- ListObjects
- ListObjectsV2

- ListObjectVersions
- ListParts
- PutObject
- PutObjectTagging
- UploadPart
- UploadPartCopy

Operaciones de la API de Amazon S3 Control para administrar buckets

S3 en Outposts admite las siguientes operaciones de la API de Amazon S3 Control para trabajar con buckets.

- 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

Operaciones de la API de S3 en Outposts para administrar Outposts

S3 en Outposts admite las siguientes operaciones de la API de Amazon S3 en Outposts para administrar puntos de conexión.

- CreateEndpoint
- DeleteEndpoint
- ListEndpoints
- Listar publicaciones salientes con S3
- ListSharedEndpoints

Configure el cliente de control de S3 para S3 en Outposts con SDK para Java

En el siguiente ejemplo, se configura el cliente de control de Amazon S3 para Amazon S3 en Outposts con AWS SDK for Java. Para utilizar este ejemplo, sustituya *user input placeholder* por su propia información.

```
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();

}
```

Realización de solicitudes a S3 en Outposts mediante IPv6

Los puntos de conexión de doble pila de Amazon S3 en Outposts y S3 en Outposts permiten realizar solicitudes a buckets de S3 en Outposts con el protocolo IPv6 o IPv4. Gracias a la compatibilidad de IPv6 con S3 en Outposts, puede acceder a sus buckets y recursos del plano de control a través de las API de S3 en Outposts mediante redes IPv6.

> **Note**
>
> Las acciones de los objetos de S3 en Outposts (como `PutObject` o `GetObject`) no son compatibles con las redes IPv6.

No hay cargos adicionales por acceder a S3 en Outposts mediante redes IPv6. Para obtener más información acerca de S3 en Outposts, consulte Precios del bastidor de AWS Outposts.

Temas

- Introducción a IPv6
- Uso de puntos de conexión de doble pila para realizar solicitudes mediante una red IPv6
- Uso de direcciones IPv6 en políticas de IAM
- Probar la compatibilidad de dirección IP
- Uso de IPv6 con AWS PrivateLink
- Uso de puntos de conexión de doble pila en S3 en Outposts

Introducción a IPv6

Para realizar una solicitud a un bucket de S3 en Outposts mediante IPv6, debe utilizar un punto de conexión de doble pila. En la siguiente sección se describe cómo hacer solicitudes mediante IPv6 con los puntos de enlace de doble pila.

A continuación se describen algunos puntos importantes a tener en cuenta antes de acceder a un bucket de S3 en Outposts mediante IPv6:

- El cliente y la red que acceden al bucket deben estar autorizados para utilizar IPv6.

- Se admiten tanto solicitudes de estilo alojamiento virtual como de tipo ruta para el acceso a IPv6. Para obtener más información, consulte Uso de puntos de conexión de doble pila en S3 en Outposts.
- Si utiliza el filtrado de direcciones IP de origen en sus políticas de bucket de S3 en Outposts o de usuario de AWS Identity and Access Management (IAM), debe actualizar las políticas para que incluyan los rangos de direcciones IPv6.

> **Note**
>
> Este requisito solo se aplica a las operaciones de buckets de S3 en Outposts y a los recursos del plano de control en redes IPv6. Las acciones de los objetos de Amazon S3 en Outposts no son compatibles con las redes IPv6.

- Cuando utiliza IPv6, los archivos de registro de acceso al servidor producen direcciones IP en un formato de IPv6. Debe actualizar el software, las herramientas y los scripts existentes que utiliza para analizar archivos de registro de S3 en Outposts para que puedan analizar las direcciones IP remotas con formato IPv6. A continuación, las herramientas, los scripts y el software actualizados analizarán correctamente las direcciones IP remotas con formato IPv6.

Uso de puntos de conexión de doble pila para realizar solicitudes mediante una red IPv6

Para realizar solicitudes con llamadas a la API de S3 en Outposts a través de IPv6, puede usar puntos de conexión de doble pila mediante la AWS CLI o el SDK de AWS. Las operaciones de la API de Amazon S3 Control para administrar buckets y las operaciones de la API de S3 en Outposts para administrar Outposts funcionan igual tanto si se accede a S3 en Outposts a través de un protocolo IPv6 como de un protocolo IPv4. Sin embargo, debe de tener en cuenta que las operaciones de la API de Amazon S3 en Outposts (como `PutObject` o `GetObject`) no son compatibles con las redes IPv6.

Al usar AWS Command Line Interface (AWS CLI) y los SDK de AWS, puede utilizar un parámetro o una marca para cambiar a un punto de enlace de doble pila. También puede especificar el punto de conexión de doble pila directamente como una anulación del punto de conexión de S3 en Outposts en el archivo de configuración.

Puede utilizar un punto de conexión de doble pila para acceder a un bucket de S3 en Outposts mediante IPv6 desde cualquiera de las siguientes opciones:

- La AWS CLI, consulte Usar puntos de enlace de doble pila desde la AWS CLI.
- Para los SDK de AWS, consulte Uso de los puntos de conexión de doble pila de S3 en Outposts desde los SDK de AWS.

Uso de direcciones IPv6 en políticas de IAM

Antes de intentar acceder a un bucket de S3 en Outposts mediante un protocolo IPv6, debe asegurarse de que los usuarios de IAM o las políticas de bucket de S3 en Outposts utilizadas para el filtrado de direcciones IP estén actualizadas e incluyan los rangos de direcciones IPv6. Si las políticas de filtrado de direcciones IP no están actualizadas para gestionar direcciones IPv6, puede perder el acceso a un bucket de S3 en Outposts al intentar usar el protocolo IPv6.

Las políticas de IAM que filtran direcciones IP utilizan operadores de condición de dirección IP. La siguiente política de buckets de S3 en Outposts identifica el rango IP 54.240.143.* de las direcciones IPv4 permitidas con operadores de condición de dirección IP. Cualquier dirección IP fuera de este rango no podrá acceder al bucket de S3 en Outposts (`DOC-EXAMPLE-BUCKET`). Dado que todas las direcciones IPv6 están fuera del rango permitido, esta política evita que las direcciones IPv6 puedan acceder a `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"}
      }
    }
  ]
}
```

Puede modificar el elemento `Condition` de la política de bucket de S3 en Outposts para permitir los rangos de direcciones IPv4 (`54.240.143.0/24`) e IPv6 (`2001:DB8:1234:5678::/64`), tal

como se muestra en el siguiente ejemplo. Puede utilizar el mismo tipo de bloque `Condition` que se muestra en el ejemplo para actualizar las políticas de bucket y de usuario de IAM.

```
       "Condition": {
         "IpAddress": {
            "aws:SourceIp": [
              "54.240.143.0/24",
               "2001:DB8:1234:5678::/64"
             ]
          }
        }
```

Antes de utilizar IPv6, debe actualizar todas las políticas de bucket y de usuario de IAM; relevantes que utilizan filtrado de direcciones IP para permitir los rangos de direcciones IPv6. Le recomendamos que actualice sus políticas de IAM con los rangos de direcciones IPv6 de la organización además de los rangos de direcciones IPv4 existentes. Para ver un ejemplo de una política de bucket que permite el acceso a través de IPv6 e IPv4, consulte Restringir el acceso a direcciones IP específicas.

Puede revisar sus políticas de usuario de IAM en la consola de IAM en https://console.aws.amazon.com/iam/. Para obtener más información acerca de IAM, consulte la guía del usuario de IAM. Para obtener información sobre las políticas de buckets de S3 en Outposts, consulte Adición o edición de una política de bucket para un bucket de Amazon S3 en Outposts.

Probar la compatibilidad de dirección IP

Si utiliza una instancia de Linux, de Unix o una plataforma macOS X, puede probar el acceso a un punto de conexión de doble pila mediante IPv6. Por ejemplo, para probar la conexión a Amazon S3 en Outposts en los puntos de conexión mediante IPv6, utilice el comando `dig`:

```
dig s3-outposts.us-west-2.api.aws AAAA +short
```

Si el punto de conexión de doble pila a través de una red IPv6 está configurado correctamente, el comando `dig` devuelve las direcciones IPv6 conectadas. Por ejemplo:

```
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
```

Uso de IPv6 con AWS PrivateLink

S3 en Outposts admite el protocolo IPv6 para los servicios y puntos de conexión de AWS PrivateLink. Gracias a la compatibilidad de AWS PrivateLink con el protocolo IPv6, puede conectarse a los puntos de conexión de servicio de su VPC a través de redes IPv6, ya sea desde conexiones en las instalaciones o desde otras conexiones privadas. La compatibilidad de IPv6 con AWS PrivateLink para S3 en Outposts también le permite integrar AWS PrivateLink con puntos de conexión de doble pila. Para ver los pasos a seguir para habilitar IPv6 para AWS PrivateLink, consulte Expedite your IPv6 adoption with AWS PrivateLink services and endpoints.

> **Note**
>
> Para actualizar el tipo de dirección IP compatible de IPv4 a IPv6, consulte Modify the supported IP address type en la Guía del usuario de AWS PrivateLink.

Uso de IPv6 con AWS PrivateLink

Si usa AWS PrivateLink con IPv6, debe crear un punto de conexión de interfaz de VPC de doble pila o de IPv6. Para ver los pasos generales a seguir para crear un punto de conexión de VPC desde la AWS Management Console, consulte Access an AWS service using an interface VPC endpoint en la Guía del usuario de AWS PrivateLink.

AWS Management Console

Utilice el siguiente procedimiento para crear un punto de conexión de VPC de interfaz que se conecte a S3 en Outposts.

1. Inicie sesión en la AWS Management Console y abra la consola de VPC en https://console.aws.amazon.com/vpc/.
2. En el panel de navegación, elija Puntos de conexión.
3. Elija Crear punto de conexión.
4. En Service category (Categoría de servicios), elija AWSServices (Servicios de AWC).
5. En Nombre del servicio, elija el servicio S3 en Outposts (com.amazonaws.us-east-1.s3-outposts).
6. En VPC, elija la VPC desde la que accederá a S3 en Outposts.

7. En Subredes, seleccione una subred por zona de disponibilidad desde la que accederá a S3 en Outposts. No puede seleccionar varias subredes de la misma zona de disponibilidad. Por cada subred que seleccione, se creará una interfaz de red de punto de conexión nueva. De forma predeterminada, las direcciones IP de los rangos de direcciones IP de la subred se asignan a las interfaces de red de los puntos de conexión. Para designar una dirección IP para una interfaz de red de puntos de conexión, elija Designar direcciones IP e introduzca una dirección IPv6 del rango de direcciones de la subred.
8. Para Tipo de dirección IP, elija Dualstack. Asigne ambas direcciones IPv4 e IPv6 a sus interfaces de red del punto de conexión. Esta opción solo se admite si todas las subredes seleccionadas tienen rangos de direcciones IPv4 e IPv6.
9. En Grupos de seguridad, elija los grupos de seguridad para asociarlos a las interfaces de red del punto de conexión para el punto de conexión de VPC. De forma predeterminada, el grupo de seguridad predeterminado está asociado a la VPC.
10. En Política, elija Acceso completo para permitir todas las operaciones de todas las entidades principales en todos los recursos del punto de conexión de VPC. De lo contrario, elija Personalizar para adjuntar una política de punto de conexión de VPC que controle los permisos que tienen las entidades principales para realizar acciones en los recursos a través del punto de conexión de VPC. Esta opción solo está disponible si el servicio admite las políticas de punto de conexión de VPC. Para obtener más información, consulte Endpoint policies.
11. (Opcional) Para agregar una etiqueta, elija Agregar etiqueta nueva e ingrese la clave y el valor de la etiqueta.
12. Seleccione Crear punto de conexión.

Example Ejemplo de política de bucket de S3 en Outposts

Para permitir que S3 en Outposts interactúe con sus puntos de conexión de VPC, puede actualizar la política de S3 en Outposts de la siguiente manera:

```
{
    "Statement": [
        {
            "Effect": "Allow",
            "Action": "s3-outposts:*",
            "Resource": "*",
            "Principal": "*"
        }
```

```
    ]
}
```

AWS CLI

> **Note**
>
> Para habilitar la red IPv6 en su punto de conexión de VPC, debe configurar `IPv6` para el filtro `SupportedIpAddressType` para S3 en Outposts.

En el siguiente ejemplo se utiliza el comando `create-vpc-endpoint` para crear un nuevo punto de conexión de interfaz de doble pila.

```
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"
```

Según la configuración del servicio de AWS PrivateLink, es posible que el proveedor de servicios de puntos de conexión de VPC deba aceptar las conexiones de punto de conexión recién creadas antes de utilizarlas. Para obtener más información, consulte Accept and reject endpoint connection requests en la Guía del usuario de AWS PrivateLink.

En el siguiente ejemplo, se usa el comando `modify-vpc-endpoint` para actualizar el punto de conexión de VPC solo para IPV por un punto de conexión de doble pila. El punto de conexión de doble pila permite acceder a las redes IPv4 e 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"
```

Para ver más información sobre cómo habilitar la red IPv6 para AWS PrivateLink, consulte la publicación Expedite your IPv6 adoption with AWS PrivateLink services and endpoints.

Uso de puntos de conexión de doble pila en S3 en Outposts

Los puntos de conexión de doble pila de S3 en Outposts permiten realizar solicitudes a los buckets de S3 en Outposts a través de IPv6 y de IPv4. En esta sección se describe cómo utilizar los puntos de conexión de doble pila de S3 en Outposts.

Temas

Puntos de conexión de doble pila de S3 en Outposts

Cuando realiza una solicitud a un punto de conexión de doble pila, la URL del bucket de S3 en Outposts resulta en una dirección IPv6 o IPv4. Para obtener más información acerca de un bucket de S3 en Outposts mediante IPv6, consulte Realización de solicitudes a S3 en Outposts mediante IPv6.

Para obtener acceso a un bucket de S3 en Outposts mediante un punto de conexión de doble pila, use un nombre de punto de conexión de tipo ruta. S3 en Outposts solo admite nombres de puntos de conexión de doble pila regionales, por lo que debe especificar la región dentro del nombre.

Los puntos de conexión FIPS de doble pila de tipo ruta utilizan la siguiente convención de nomenclatura:

```
s3-outposts-fips.region.api.aws
```

Los puntos de conexión FIPS que no son de doble pila utilizan la siguiente convención de nomenclatura:

```
s3-outposts.region.api.aws
```

> **Note**
>
> Los nombres de punto de conexión de tipo de alojamiento virtual no son compatibles con S3 en Outposts.

Usar puntos de enlace de doble pila desde la AWS CLI

Esta sección proporciona ejemplos de comandos de la AWS CLI. que se usan para realizar solicitudes a un punto de conexión de doble pila. Para obtener instrucciones acerca de cómo configurar la AWS CLI, consulte Introducción mediante AWS CLI y SDK para Java.

Puede establecer el valor de configuración `use_dualstack_endpoint` en `true` en un perfil de su archivo de AWS Config para dirigir todas las solicitudes de Amazon S3 que realicen los comandos `s3` y `s3api` de la AWS CLI al punto de conexión de doble pila para la región especificada. Puede especificar la región en el archivo de configuración o en un comando utilizando la opción `--region`.

Si utiliza puntos de conexión de doble pila con la AWS CLI, solo se admiten los estilos de direccionamiento `path`. El estilo de direccionamiento configurado en el archivo de configuración determina si el nombre del bucket está en el name de host o en la URL. Para obtener más información, consulte s3outposts en la Guía del usuario de AWS CLI.

Para usar un punto de conexión de doble pila mediante la AWS CLI, utilice el parámetro `--endpoint-url` junto con el punto de conexión `http://s3.dualstack.`*`region`*`.amazonaws.com` o `https://s3-outposts-fips.`*`region`*`.api.aws` para cualquiera de los comandos `s3control` o `s3outposts`.

Por ejemplo:

```
$  aws s3control list-regional-buckets --endpoint-url https://s3-
outposts.region.api.aws
```

Uso de los puntos de conexión de doble pila de S3 en Outposts desde los SDK de AWS

En esta sección, se proporcionan ejemplos de cómo obtener acceso a un punto de enlace de doble pila con los SDK de AWS.

AWS SDK for Java 2.xEjemplo de punto de enlace de doble pila con

En el siguiente ejemplo se muestra cómo usar las clases `S3ControlClient` y `S3OutpostsClient` para habilitar puntos de conexión de doble pila al crear un cliente S3 en Outposts con AWS SDK for Java 2.x. Para obtener instrucciones sobre cómo crear y probar un ejemplo de Java funcional para Amazon S3 en Outposts, consulte Introducción mediante AWS CLI y SDK para Java.

Example — Crear una clase de **`S3ControlClient`** con los puntos de conexión de doble pila habilitados

```
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 — Crear un **`S3OutpostsClient`** con los puntos de conexión de doble pila habilitados

```
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 utiliza AWS SDK for Java 2.x en Windows, es probable que tenga que configurar adecuadamente la siguiente propiedad de la máquina virtual Java (JVM):

```
java.net.preferIPv6Addresses=true
```

Ejemplos de código de Amazon S3 con SDK de AWS

Los siguientes ejemplos de código muestran cómo utilizar Amazon S3 con un kit de desarrollo de software (SDK) de AWS.

Las acciones son extractos de código de programas más grandes y deben ejecutarse en contexto. Mientras las acciones muestran cómo llamar a las funciones de servicio individuales, es posible ver las acciones en contexto en los escenarios relacionados y en los ejemplos entre servicios.

Los escenarios son ejemplos de código que muestran cómo llevar a cabo una tarea específica llamando a varias funciones dentro del mismo servicio.

Los ejemplos con varios servicios son aplicaciones de muestra que funcionan con varios Servicios de AWS.

Para obtener una lista completa de las guías para desarrolladores del AWS SDK y ejemplos de código, consulte Uso de este servicio con un SDK de AWS. En este tema también se incluye información sobre cómo comenzar a utilizar el SDK y detalles sobre sus versiones anteriores.

Introducción

Introducción a Amazon S3

En los siguientes ejemplos de código se muestra cómo empezar a utilizar Amazon S3.

C++

SDK para C++

> **Note**
>
> Hay más en GitHub. Busque el ejemplo completo y aprenda a configurar y ejecutar en el Repositorio de ejemplos de código de AWS.

Código del archivo de CMake CMakeLists.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})
```

Código del archivo de origen 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;
}
```

- Para obtener información sobre la API, consulte ListBuckets en la Referencia de la API de AWS SDK for C++.

Go

SDK para Go V2

> **Note**
>
> Hay más en GitHub. Busque el ejemplo completo y aprenda a configurar y ejecutar en el Repositorio de ejemplos de código de 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)
  }
 }
}
```

- Para obtener información sobre la API, consulte ListBuckets en la Referencia de la API de AWS SDK for Go.

Java

SDK para Java 2.x

> **Note**
>
> Hay más en GitHub. Busque el ejemplo completo y aprenda a configurar y ejecutar en el Repositorio de ejemplos de código de 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);
        }
    }
}
```

- Para obtener información sobre la API, consulte ListBuckets en la Referencia de la API de AWS SDK for Java 2.x.

JavaScript

SDK para JavaScript (v3)

> **Note**
>
> Hay más información en GitHub. Busque el ejemplo completo y aprenda a configurar y ejecutar en el Repositorio de ejemplos de código de 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;
};
```

- Para obtener información sobre la API, consulte ListBuckets en la Referencia de la API de AWS SDK for JavaScript.

PHP

SDK para PHP

> **Note**
>
> Hay más en GitHub. Busque el ejemplo completo y aprenda a configurar y ejecutar en el Repositorio de ejemplos de código de AWS.

```
use Aws\S3\S3Client;

$client = new S3Client(['region' => 'us-west-2']);
$results = $client->listBuckets();
var_dump($results);
```

- Para obtener información sobre la API, consulte ListBuckets en la Referencia de la API de AWS SDK for PHP.

Python

SDK para Python (Boto3)

> **Note**
>
> Hay más en GitHub. Busque el ejemplo completo y aprenda a configurar y ejecutar en el Repositorio de ejemplos de código de 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()
```

- Para obtener información sobre la API, consulte ListBuckets en la Referencia de la API de AWS SDK para Python (Boto3).

Ruby

SDK para Ruby

> **Note**
>
> Hay más en GitHub. Busque el ejemplo completo y aprenda a configurar y ejecutar en el Repositorio de ejemplos de código de AWS.

```
# frozen_string_literal: true

# S3Manager is a class responsible for managing S3 operations
# such as listing all S3 buckets in the current AWS account.
class S3Manager
  def initialize(client)
    @client = client
    @logger = Logger.new($stdout)
  end

  # Lists and prints all S3 buckets in the current AWS account.
  def list_buckets
    @logger.info('Here are the buckets in your account:')

    response = @client.list_buckets
```

```
    if response.buckets.empty?
      @logger.info("You don't have any S3 buckets yet.")
    else
      response.buckets.each do |bucket|
        @logger.info("- #{bucket.name}")
      end
    end
  rescue Aws::Errors::ServiceError => e
    @logger.error("Encountered an error while listing buckets: #{e.message}")
  end
end

if $PROGRAM_NAME == __FILE__
  s3_client = Aws::S3::Client.new
  manager = S3Manager.new(s3_client)
  manager.list_buckets
end
```

- Para obtener información sobre la API, consulte ListBuckets en la Referencia de la API de AWS SDK for Ruby.

Ejemplos de código

- Acciones de Amazon S3 con SDK de AWS
  - Uso de AbortMultipartUpload con un AWS SDK o la CLI
  - Uso de AbortMultipartUploads con un AWS SDK o la CLI
  - Uso de CompleteMultipartUpload con un AWS SDK o la CLI
  - Uso de CopyObject con un AWS SDK o la CLI
  - Uso de CreateBucket con un AWS SDK o la CLI
  - Uso de CreateMultiRegionAccessPoint con un AWS SDK o la CLI
  - Uso de CreateMultipartUpload con un AWS SDK o la CLI
  - Uso de DeleteBucket con un AWS SDK o la CLI
  - Uso de DeleteBucketAnalyticsConfiguration con un AWS SDK o la CLI
  - Uso de DeleteBucketCors con un AWS SDK o la CLI
  - Uso de DeleteBucketEncryption con un AWS SDK o la CLI
  - Uso de DeleteBucketInventoryConfiguration con un AWS SDK o la CLI

- Uso de DeleteBucketLifecycle con un AWS SDK o la CLI
- Uso de DeleteBucketMetricsConfiguration con un AWS SDK o la CLI
- Uso de DeleteBucketPolicy con un AWS SDK o la CLI
- Uso de DeleteBucketReplication con un AWS SDK o la CLI
- Uso de DeleteBucketTagging con un AWS SDK o la CLI
- Uso de DeleteBucketWebsite con un AWS SDK o la CLI
- Uso de DeleteObject con un AWS SDK o la CLI
- Uso de DeleteObjectTagging con un AWS SDK o la CLI
- Uso de DeleteObjects con un AWS SDK o la CLI
- Uso de DeletePublicAccessBlock con un AWS SDK o la CLI
- Uso de GetBucketAccelerateConfiguration con un AWS SDK o la CLI
- Uso de GetBucketAcl con un AWS SDK o la CLI
- Uso de GetBucketAnalyticsConfiguration con un AWS SDK o la CLI
- Uso de GetBucketCors con un AWS SDK o la CLI
- Uso de GetBucketEncryption con un AWS SDK o la CLI
- Uso de GetBucketInventoryConfiguration con un AWS SDK o la CLI
- Uso de GetBucketLifecycleConfiguration con un AWS SDK o la CLI
- Uso de GetBucketLocation con un AWS SDK o la CLI
- Uso de GetBucketLogging con un AWS SDK o la CLI
- Uso de GetBucketMetricsConfiguration con un AWS SDK o la CLI
- Uso de GetBucketNotification con un AWS SDK o la CLI
- Uso de GetBucketPolicy con un AWS SDK o la CLI
- Uso de GetBucketPolicyStatus con un AWS SDK o la CLI
- Uso de GetBucketReplication con un AWS SDK o la CLI
- Uso de GetBucketRequestPayment con un AWS SDK o la CLI
- Uso de GetBucketTagging con un AWS SDK o la CLI
- Uso de GetBucketVersioning con un AWS SDK o la CLI
- Uso de GetBucketWebsite con un AWS SDK o la CLI
- Uso de GetObject con un AWS SDK o la CLI
- Uso de GetObjectAcl con un AWS SDK o la CLI

- Uso de GetObjectAttributes con un AWS SDK o la CLI
- Uso de GetObjectLegalHold con un AWS SDK o la CLI
- Uso de GetObjectLockConfiguration con un AWS SDK o la CLI
- Uso de GetObjectRetention con un AWS SDK o la CLI
- Uso de GetObjectTagging con un AWS SDK o la CLI
- Uso de GetPublicAccessBlock con un AWS SDK o la CLI
- Uso de HeadBucket con un AWS SDK o la CLI
- Uso de HeadObject con un AWS SDK o la CLI
- Uso de ListBucketAnalyticsConfigurations con un AWS SDK o la CLI
- Uso de ListBucketInventoryConfigurations con un AWS SDK o la CLI
- Uso de ListBuckets con un AWS SDK o la CLI
- Uso de ListMultipartUploads con un AWS SDK o la CLI
- Uso de ListObjectVersions con un AWS SDK o la CLI
- Uso de ListObjects con un AWS SDK o la CLI
- Uso de ListObjectsV2 con un AWS SDK o la CLI
- Uso de PutBucketAccelerateConfiguration con un AWS SDK o la CLI
- Uso de PutBucketAcl con un AWS SDK o la CLI
- Uso de PutBucketCors con un AWS SDK o la CLI
- Uso de PutBucketEncryption con un AWS SDK o la CLI
- Uso de PutBucketLifecycleConfiguration con un AWS SDK o la CLI
- Uso de PutBucketLogging con un AWS SDK o la CLI
- Uso de PutBucketNotification con un AWS SDK o la CLI
- Uso de PutBucketNotificationConfiguration con un AWS SDK o la CLI
- Uso de PutBucketPolicy con un AWS SDK o la CLI
- Uso de PutBucketReplication con un AWS SDK o la CLI
- Uso de PutBucketRequestPayment con un AWS SDK o la CLI
- Uso de PutBucketTagging con un AWS SDK o la CLI
- Uso de PutBucketVersioning con un AWS SDK o la CLI
- Uso de PutBucketWebsite con un AWS SDK o la CLI
- Uso de PutObject con un AWS SDK o la CLI

- Uso de PutObjectAcl con un AWS SDK o la CLI
- Uso de PutObjectLegalHold con un AWS SDK o la CLI
- Uso de PutObjectLockConfiguration con un AWS SDK o la CLI
- Uso de PutObjectRetention con un AWS SDK o la CLI
- Uso de RestoreObject con un AWS SDK o la CLI
- Uso de SelectObjectContent con un AWS SDK o la CLI
- Uso de UploadPart con un AWS SDK o la CLI

- Escenarios de Amazon S3 con SDK de AWS
  - Cree una URL prefirmada para Amazon S3 mediante un SDK de AWS
  - Una página web que indica los objetos de Amazon S3 que usan un SDK de AWS
  - Eliminación de las cargas multiparte incompletas a Amazon S3 mediante un AWS SDK
  - Descargar todos los objetos de un bucket de Amazon Simple Storage Service (Amazon S3) en un directorio local
  - Obtención de un objeto de Amazon S3 desde un punto de acceso de varias regiones con un SDK de AWS
  - Obtenga un objeto de un bucket de Amazon S3 con un SDK de AWS al especificar un encabezado If-Modified-Since
  - Introducción a los buckets y objetos de Amazon S3 con un SDK de AWS
  - Introducción al cifrado de objetos de Amazon S3 con un SDK de AWS
  - Introducción a etiquetas de objetos de Amazon S3 con un SDK de AWS
  - Obtención de la configuración de retención legal de un objeto de Amazon S3 mediante un SDK de AWS
  - Trabajo con las características de bloqueo de objetos de Amazon S3 mediante un SDK de AWS
  - Administre listas de control de acceso (ACL) para buckets de Amazon S3 con un SDK de AWS
  - Administre objetos de Amazon S3 con control de versiones en lotes con una función de Lambda mediante un SDK de AWS
  - Analizar los URI de Amazon S3 mediante un SDK de AWS
  - Ejecución de una copia multiparte de un objeto de Amazon S3 con un SDK de AWS
  - Ejecución de una carga multiparte de un objeto de Amazon S3 con un AWS SDK
  - Reciba y procese las notificaciones de eventos de Amazon S3 mediante un AWS SDK.
  - Envío de notificaciones de eventos de S3 a Amazon EventBridge mediante un AWS SDK

- Realización de un seguimiento de la carga o descarga de un objeto de Amazon S3 mediante un AWS SDK
- Ejemplos de enfoques para pruebas unitarias y de integración con un SDK de AWS
- Cargar de forma recursiva un directorio local en un bucket de Amazon Simple Storage Service (Amazon S3)
- Cargar o descargar archivos de gran tamaño desde y hacia Amazon S3 con un SDK de AWS
- Carga de un flujo de tamaño desconocido en un objeto de Amazon S3 mediante un SDK de AWS
- Uso de sumas de comprobación para trabajar con un objeto de Amazon S3 con un SDK de AWS
- Trabajo con las características de integridad de objetos de Amazon S3 utilizando un AWS SDK
- Trabajo con objetos con control de versiones de Amazon S3 con un SDK de AWS

- Ejemplos sin servidor para Amazon S3 que utilizan SDK de AWS
  - Invocación de una función de Lambda desde un desencadenador de Amazon S3
- Ejemplos de servicios combinados de Amazon S3 con SDK de AWS
  - Cree una aplicación Amazon Transcribe
  - Convierta texto en voz y de nuevo a texto con un SDK de AWS
  - Creación de una aplicación de administración de activos fotográficos que permita a los usuarios administrar las fotos mediante etiquetas
  - Creación de una aplicación de exploración de Amazon Textract
  - Detección de EPI en imágenes con Amazon Rekognition mediante un AWS SDK
  - Detecte entidades en el texto extraído de una imagen con un SDK de AWS
  - Detecte rostros en una imagen con un SDK de AWS
  - Detección de personas y objetos en un vídeo con Amazon Rekognition mediante un AWS SDK
  - Detecte personas y objetos en un vídeo con Amazon Rekognition mediante un SDK de AWS
  - Guarde EXIF y otra información de la imagen con un SDK de AWS
  - Transformación de datos para su aplicación con S3 Object Lambda

Acciones de Amazon S3 con SDK de AWS

Los siguientes ejemplos de código muestran cómo llevar a cabo acciones individuales de Amazon S3 con los SDK de AWS. Estos fragmentos llaman a la API de Amazon S3 y son fragmentos de código

de programas más grandes que deben ejecutarse en contexto. En cada ejemplo se incluye un enlace a GitHub, con instrucciones de configuración y ejecución del código.

Los siguientes ejemplos incluyen solo las acciones que se utilizan con mayor frecuencia. Para obtener una lista completa, consulte la Referencia de la API de Amazon Simple Storage Service (Amazon S3).

Ejemplos

- Uso de AbortMultipartUpload con un AWS SDK o la CLI
- Uso de AbortMultipartUploads con un AWS SDK o la CLI
- Uso de CompleteMultipartUpload con un AWS SDK o la CLI
- Uso de CopyObject con un AWS SDK o la CLI
- Uso de CreateBucket con un AWS SDK o la CLI
- Uso de CreateMultiRegionAccessPoint con un AWS SDK o la CLI
- Uso de CreateMultipartUpload con un AWS SDK o la CLI
- Uso de DeleteBucket con un AWS SDK o la CLI
- Uso de DeleteBucketAnalyticsConfiguration con un AWS SDK o la CLI
- Uso de DeleteBucketCors con un AWS SDK o la CLI
- Uso de DeleteBucketEncryption con un AWS SDK o la CLI
- Uso de DeleteBucketInventoryConfiguration con un AWS SDK o la CLI
- Uso de DeleteBucketLifecycle con un AWS SDK o la CLI
- Uso de DeleteBucketMetricsConfiguration con un AWS SDK o la CLI
- Uso de DeleteBucketPolicy con un AWS SDK o la CLI
- Uso de DeleteBucketReplication con un AWS SDK o la CLI
- Uso de DeleteBucketTagging con un AWS SDK o la CLI
- Uso de DeleteBucketWebsite con un AWS SDK o la CLI
- Uso de DeleteObject con un AWS SDK o la CLI
- Uso de DeleteObjectTagging con un AWS SDK o la CLI
- Uso de DeleteObjects con un AWS SDK o la CLI
- Uso de DeletePublicAccessBlock con un AWS SDK o la CLI
- Uso de GetBucketAccelerateConfiguration con un AWS SDK o la CLI
- Uso de GetBucketAcl con un AWS SDK o la CLI

- Uso de GetBucketAnalyticsConfiguration con un AWS SDK o la CLI
- Uso de GetBucketCors con un AWS SDK o la CLI
- Uso de GetBucketEncryption con un AWS SDK o la CLI
- Uso de GetBucketInventoryConfiguration con un AWS SDK o la CLI
- Uso de GetBucketLifecycleConfiguration con un AWS SDK o la CLI
- Uso de GetBucketLocation con un AWS SDK o la CLI
- Uso de GetBucketLogging con un AWS SDK o la CLI
- Uso de GetBucketMetricsConfiguration con un AWS SDK o la CLI
- Uso de GetBucketNotification con un AWS SDK o la CLI
- Uso de GetBucketPolicy con un AWS SDK o la CLI
- Uso de GetBucketPolicyStatus con un AWS SDK o la CLI
- Uso de GetBucketReplication con un AWS SDK o la CLI
- Uso de GetBucketRequestPayment con un AWS SDK o la CLI
- Uso de GetBucketTagging con un AWS SDK o la CLI
- Uso de GetBucketVersioning con un AWS SDK o la CLI
- Uso de GetBucketWebsite con un AWS SDK o la CLI
- Uso de GetObject con un AWS SDK o la CLI
- Uso de GetObjectAcl con un AWS SDK o la CLI
- Uso de GetObjectAttributes con un AWS SDK o la CLI
- Uso de GetObjectLegalHold con un AWS SDK o la CLI
- Uso de GetObjectLockConfiguration con un AWS SDK o la CLI
- Uso de GetObjectRetention con un AWS SDK o la CLI
- Uso de GetObjectTagging con un AWS SDK o la CLI
- Uso de GetPublicAccessBlock con un AWS SDK o la CLI
- Uso de HeadBucket con un AWS SDK o la CLI
- Uso de HeadObject con un AWS SDK o la CLI
- Uso de ListBucketAnalyticsConfigurations con un AWS SDK o la CLI
- Uso de ListBucketInventoryConfigurations con un AWS SDK o la CLI
- Uso de ListBuckets con un AWS SDK o la CLI
- Uso de ListMultipartUploads con un AWS SDK o la CLI

- Uso de ListObjectVersions con un AWS SDK o la CLI
- Uso de ListObjects con un AWS SDK o la CLI
- Uso de ListObjectsV2 con un AWS SDK o la CLI
- Uso de PutBucketAccelerateConfiguration con un AWS SDK o la CLI
- Uso de PutBucketAcl con un AWS SDK o la CLI
- Uso de PutBucketCors con un AWS SDK o la CLI
- Uso de PutBucketEncryption con un AWS SDK o la CLI
- Uso de PutBucketLifecycleConfiguration con un AWS SDK o la CLI
- Uso de PutBucketLogging con un AWS SDK o la CLI
- Uso de PutBucketNotification con un AWS SDK o la CLI
- Uso de PutBucketNotificationConfiguration con un AWS SDK o la CLI
- Uso de PutBucketPolicy con un AWS SDK o la CLI
- Uso de PutBucketReplication con un AWS SDK o la CLI
- Uso de PutBucketRequestPayment con un AWS SDK o la CLI
- Uso de PutBucketTagging con un AWS SDK o la CLI
- Uso de PutBucketVersioning con un AWS SDK o la CLI
- Uso de PutBucketWebsite con un AWS SDK o la CLI
- Uso de PutObject con un AWS SDK o la CLI
- Uso de PutObjectAcl con un AWS SDK o la CLI
- Uso de PutObjectLegalHold con un AWS SDK o la CLI
- Uso de PutObjectLockConfiguration con un AWS SDK o la CLI
- Uso de PutObjectRetention con un AWS SDK o la CLI
- Uso de RestoreObject con un AWS SDK o la CLI
- Uso de SelectObjectContent con un AWS SDK o la CLI
- Uso de UploadPart con un AWS SDK o la CLI

Uso de **`AbortMultipartUpload`** con un AWS SDK o la CLI

Los siguientes ejemplos de código muestran cómo utilizar `AbortMultipartUpload`.

Los ejemplos de acciones son extractos de código de programas más grandes y deben ejecutarse en contexto. Puede ver esta acción en su contexto en los siguientes ejemplos de código:

- Eliminación de cargas multiparte incompletas
- Trabajo con la integridad de los objetos de Amazon S3

C++

SDK para C++

> **Note**
>
> Hay más en GitHub. Busque el ejemplo completo y aprenda a configurar y ejecutar en el Repositorio de ejemplos de código de AWS.

```
//! Abort a multipart upload to an S3 bucket.
/*!
    \param bucket: The name of the S3 bucket where the object will be uploaded.
    \param key: The unique identifier (key) for the object within the S3 bucket.
    \param uploadID: An upload ID string.
    \param client: The S3 client instance used to perform the upload operation.
    \return bool: Function succeeded.
*/

bool AwsDoc::S3::abortMultipartUpload(const Aws::String &bucket,
                                      const Aws::String &key,
                                      const Aws::String &uploadID,
                                      const Aws::S3::S3Client &client) {
    Aws::S3::Model::AbortMultipartUploadRequest request;
    request.SetBucket(bucket);
    request.SetKey(key);
    request.SetUploadId(uploadID);

    Aws::S3::Model::AbortMultipartUploadOutcome outcome =
            client.AbortMultipartUpload(request);

    if (outcome.IsSuccess()) {
        std::cout << "Multipart upload aborted." << std::endl;
    } else {
        std::cerr << "Error aborting multipart upload: " <<
 outcome.GetError().GetMessage() << std::endl;
    }
```

```
    return outcome.IsSuccess();
}
```

- Para obtener información sobre la API, consulte AbortMultipartUpload en la Referencia de la API de AWS SDK for C++.

CLI

AWS CLI

Anular la carga multiparte especificada

El siguiente comando `abort-multipart-upload` anula una carga multiparte de la clave `multipart/01` en el bucket `my-bucket`.

```
aws s3api abort-multipart-upload \
    --bucket my-bucket \
    --key multipart/01 \
    --upload-
id dfRtDYU0WWCCcH43C3WFbkRONycyCpTJJvxu2i5GYkZljF.Yxwh6XG7WfS2vC4to6HiV6Yjlx.cph0gtNBtJ8P
```

El ID de carga requerido por este comando se genera mediante `create-multipart-upload` y también se puede recuperar con `list-multipart-uploads`.

- Para obtener información sobre la API, consulte AbortMultipartUploads en la Referencia de comandos de la AWS CLI.

PowerShell

Herramientas para PowerShell

Ejemplo 1: este comando anula las cargas multiparte creadas hace menos de 5 días.

```
Remove-S3MultipartUpload -BucketName test-files -DaysBefore 5
```

Ejemplo 2: este comando anula las cargas multiparte creadas antes del 2 de enero de 2014.

```
Remove-S3MultipartUpload -BucketName test-files -InitiatedDate "Thursday, January
 02, 2014"
```

Ejemplo 3: este comando anula las cargas multiparte creadas antes del 2 de enero de 2014 a las 10:45:37.

```
Remove-S3MultipartUpload -BucketName test-files -InitiatedDate "2014/01/02
 10:45:37"
```

- Para obtener información sobre la API, consulte AbortMultipartUploads en la Referencia de Cmdlet de AWS Tools for PowerShell.

Para obtener una lista completa de las guías para desarrolladores del AWS SDK y ejemplos de código, consulte Uso de este servicio con un SDK de AWS. En este tema también se incluye información sobre cómo comenzar a utilizar el SDK y detalles sobre sus versiones anteriores.

Uso de **AbortMultipartUploads** con un AWS SDK o la CLI

En el siguiente ejemplo de código, se muestra cómo usar `AbortMultipartUploads`.

.NET

AWS SDK for .NET

> **Note**
>
> Hay más en GitHub. Busque el ejemplo completo y aprenda a configurar y ejecutar en el Repositorio de ejemplos de código de 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}");
            }
        }
    }
```

- Para obtener información acerca de la API, consulte AbortMultipartUploads en la Referencia de la API de AWS SDK for .NET.

Para obtener una lista completa de las guías para desarrolladores del AWS SDK y ejemplos de código, consulte Uso de este servicio con un SDK de AWS. En este tema también se incluye información sobre cómo comenzar a utilizar el SDK y detalles sobre sus versiones anteriores.

Uso de **`CompleteMultipartUpload`** con un AWS SDK o la CLI

Los siguientes ejemplos de código muestran cómo utilizar `CompleteMultipartUpload`.

Los ejemplos de acciones son extractos de código de programas más grandes y deben ejecutarse en contexto. Puede ver esta acción en su contexto en los siguientes ejemplos de código:

- Ejecución de una copia multiparte
- Ejecución de una carga multiparte
- Usar sumas de comprobación
- Trabajo con la integridad de los objetos de Amazon S3

C++

SDK para C++

> **Note**
>
> Hay más en GitHub. Busque el ejemplo completo y aprenda a configurar y ejecutar en el Repositorio de ejemplos de código de AWS.

```
//! Complete a multipart upload to an S3 bucket.
/*!
    \param bucket: The name of the S3 bucket where the object will be uploaded.
    \param key: The unique identifier (key) for the object within the S3 bucket.
    \param uploadID: An upload ID string.
    \param parts: A vector of CompleteParts.
    \param client: The S3 client instance used to perform the upload operation.
    \return CompleteMultipartUploadOutcome: The request outcome.
*/
Aws::S3::Model::CompleteMultipartUploadOutcome
 AwsDoc::S3::completeMultipartUpload(const Aws::String &bucket,

  const Aws::String &key,
```

```
  const Aws::String &uploadID,

  const Aws::Vector<Aws::S3::Model::CompletedPart> &parts,

  const Aws::S3::S3Client &client) {
    Aws::S3::Model::CompletedMultipartUpload completedMultipartUpload;
    completedMultipartUpload.SetParts(parts);

    Aws::S3::Model::CompleteMultipartUploadRequest request;
    request.SetBucket(bucket);
    request.SetKey(key);
    request.SetUploadId(uploadID);
    request.SetMultipartUpload(completedMultipartUpload);

    Aws::S3::Model::CompleteMultipartUploadOutcome outcome =
            client.CompleteMultipartUpload(request);

    if (!outcome.IsSuccess()) {
        std::cerr << "Error completing multipart upload: " <<
 outcome.GetError().GetMessage() << std::endl;
    }
    return outcome;
}
```

- Para obtener información sobre la API, consulte CompleteMultipartUpload en la Referencia de la API de AWS SDK for C++.

CLI

AWS CLI

El siguiente comando completa una carga multiparte de la clave `multipart/01`en el bucket `my-bucket`:

```
aws s3api complete-multipart-upload --multipart-upload file://
mpustruct --bucket my-bucket --key 'multipart/01' --upload-
id dfRtDYU0WWCCcH43C3WFbkRONycyCpTJJvxu2i5GYkZljF.Yxwh6XG7WfS2vC4to6HiV6Yjlx.cph0gtNBtJ8P
```

El ID de carga requerido por este comando se genera mediante `create-multipart-upload` y también se puede recuperar con `list-multipart-uploads`.

La opción de carga multiparte del comando anterior utiliza una estructura JSON que describe las partes de la carga multiparte que se deben volver a ensamblar en el archivo completo. En este ejemplo, el prefijo `file://` se usa para cargar la estructura JSON desde un archivo de la carpeta local denominada `mpustruct`.

mpustruct:

```
{
  "Parts": [
    {
      "ETag": "e868e0f4719e394144ef36531ee6824c",
      "PartNumber": 1
    },
    {
      "ETag": "6bb2b12753d66fe86da4998aa33fffb0",
      "PartNumber": 2
    },
    {
      "ETag": "d0a0112e841abec9c9ec83406f0159c8",
      "PartNumber": 3
    }
  ]
}
```

El valor de ETag de cada parte que se carga aparece cada vez que se carga una parte mediante el comando `upload-part` y también se puede recuperar mediante una llamada a `list-parts` o calcularse mediante la suma de comprobación MD5 de cada parte.

Salida:

```
{
    "ETag": "\"3944a9f7a4faab7f78788ff6210f63f0-3\"",
    "Bucket": "my-bucket",
    "Location": "https://my-bucket.s3.amazonaws.com/multipart%2F01",
    "Key": "multipart/01"
}
```

- Para obtener información sobre la API, consulte CompleteMultipartUpload en la Referencia de comandos de la AWS CLI.

Rust

SDK para Rust

> **Note**
>
> Hay más información en GitHub. Busque el ejemplo completo y aprenda a configurar y ejecutar en el Repositorio de ejemplos de código de 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();
```

- Para ver los detalles de la API, consulte CompleteMultipartUpload en la Referencia de la API del SDK de AWS para Rust.

Para obtener una lista completa de las guías para desarrolladores del AWS SDK y ejemplos de código, consulte Uso de este servicio con un SDK de AWS. En este tema también se incluye información sobre cómo comenzar a utilizar el SDK y detalles sobre sus versiones anteriores.

Uso de **`CopyObject`** con un AWS SDK o la CLI

Los siguientes ejemplos de código muestran cómo utilizar `CopyObject`.

Los ejemplos de acciones son extractos de código de programas más grandes y deben ejecutarse en contexto. Puede ver esta acción en su contexto en los siguientes ejemplos de código:

- Introducción a los buckets y objetos
- Introducción al cifrado

.NET

AWS SDK for .NET

> **Note**
>
> Hay más información en GitHub. Busque el ejemplo completo y aprenda a configurar y ejecutar en el Repositorio de ejemplos de código de 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;
        }
    }
```

- Para obtener información sobre la API, consulte CopyObject en la Referencia de la API de AWS SDK for .NET.

Bash

AWS CLI con script Bash

> **Note**
>
> Hay más información en GitHub. Busque el ejemplo completo y aprenda a configurar y ejecutar en el Repositorio de ejemplos de código de 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
}
```

- Para obtener información sobre la API, consulte CopyObject en la Referencia de comandos de AWS CLI.

C++

SDK para C++

> **Note**
>
> Hay más en GitHub. Busque el ejemplo completo y aprenda a configurar y ejecutar en el Repositorio de ejemplos de código de 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();
}
```

- Para obtener información sobre la API, consulte CopyObject en la Referencia de la API de AWS SDK for C++.

CLI

AWS CLI

El siguiente comando copia un objeto de `bucket-1` a `bucket-2`:

```
aws s3api copy-object --copy-source bucket-1/test.txt --key test.txt --
bucket bucket-2
```

Salida:

```
{
    "CopyObjectResult": {
        "LastModified": "2015-11-10T01:07:25.000Z",
        "ETag": "\"589c8b79c230a6ecd5a7e1d040a9a030\""
    },
    "VersionId": "YdnYvTCVDqRRFA.NFJjy36p0hxifMlkA"
}
```

- Para obtener detalles de la API, consulte CopyObject en la Referencia de comandos de AWS CLI.

Go

SDK para Go V2

> **Note**
>
> Hay más en GitHub. Busque el ejemplo completo y aprenda a configurar y ejecutar en el Repositorio de ejemplos de código de 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
}
```

- Para obtener información sobre la API, consulte CopyObject en la Referencia de la API de AWS SDK for Go.

Java

SDK para Java 2.x

> **Note**
>
> Hay más en GitHub. Busque el ejemplo completo y aprenda a configurar y ejecutar en el Repositorio de ejemplos de código de AWS.

Copie un objeto con 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 "";
    }
```

```
}
```

Utilice un S3TransferManager para copiar un objeto de un bucket a otro. Vea el archivo completo y pruébelo.

```
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();
    }
```

- Para obtener información sobre la API, consulte CopyObject en la Referencia de la API de AWS SDK for Java 2.x.

JavaScript

SDK para JavaScript (v3)

> **Note**
>
> Hay más información en GitHub. Busque el ejemplo completo y aprenda a configurar y ejecutar en el Repositorio de ejemplos de código de AWS.

Copie el objeto.

```
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);
  }
};
```

- Para obtener información sobre la API, consulte CopyObject en la Referencia de la API de AWS SDK for JavaScript.

Kotlin

SDK para Kotlin

> **Note**
>
> Hay más información en GitHub. Busque el ejemplo completo y aprenda a configurar y ejecutar en el Repositorio de ejemplos de código de 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)
    }
}
```

- Para obtener información sobre la API, consulte CopyObject en la Referencia de la API de AWS SDK para Kotlin.

PHP

SDK para PHP

> **Note**
>
> Hay más en GitHub. Busque el ejemplo completo y aprenda a configurar y ejecutar en el Repositorio de ejemplos de código de AWS.

Copia sencilla de un objeto.

```
        $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.");
        }
```

- Para obtener información sobre la API, consulte CopyObject en la Referencia de la API de AWS SDK for PHP.

PowerShell

Herramientas para PowerShell

Ejemplo 1: este comando copia el objeto “sample.txt” del bucket “test-files” al mismo bucket, pero con la nueva clave de “sample-copy.txt”.

```
Copy-S3Object -BucketName test-files -Key sample.txt -DestinationKey sample-
copy.txt
```

Ejemplo 2: este comando copia el objeto “sample.txt” del bucket “test-files” al mismo bucket “backup-files”, pero con la nueva clave de “sample-copy.txt”.

```
Copy-S3Object -BucketName test-files -Key sample.txt -DestinationKey sample-
copy.txt -DestinationBucket backup-files
```

Ejemplo 3: este comando descarga el objeto “sample.txt” del bucket “test-files” a un archivo local con el nombre “local-sample.txt”.

```
Copy-S3Object -BucketName test-files -Key sample.txt -LocalFile local-sample.txt
```

Ejemplo 4: descarga el objeto individual en el archivo especificado. El archivo descargado se encuentra en c:\downloads\data\archive.zip

```
Copy-S3Object -BucketName test-files -Key data/archive.zip -LocalFolder c:
\downloads
```

Ejemplo 5: descarga todos los objetos que coinciden con el prefijo de clave especificado en la carpeta local. La jerarquía de claves relativa se conservará como subcarpetas en la ubicación general de descarga.

```
Copy-S3Object -BucketName test-files -KeyPrefix data -LocalFolder c:\downloads
```

- Para obtener información sobre la API, consulte CopyObject en la Referencia de Cmdlet de AWS Tools for PowerShell.

Python

SDK para Python (Boto3)

> **Note**
>
> Hay más en GitHub. Busque el ejemplo completo y aprenda a configurar y ejecutar en el Repositorio de ejemplos de código de 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
```

- Para obtener información sobre la API, consulte CopyObject en la Referencia de la API de AWS SDK para Python (Boto3).

Ruby

SDK para Ruby

> **Note**
>
> Hay más en GitHub. Busque el ejemplo completo y aprenda a configurar y ejecutar en el Repositorio de ejemplos de código de AWS.

Copie un objeto.

```
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__
```

Copie un objeto y añada cifrado del lado del servidor al objeto de destino.

```
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__
```

- Para obtener información sobre la API, consulte CopyObject en la Referencia de la API de AWS SDK for Ruby.

Rust

SDK para Rust

> **Note**
>
> Hay más información en GitHub. Busque el ejemplo completo y aprenda a configurar y ejecutar en el Repositorio de ejemplos de código de 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
}
```

- Para obtener información sobre la API, consulte CopyObject en la Referencia de la API de AWS SDK para Rust.

SAP ABAP

SDK de SAP ABAP

> **Note**
>
> Hay más en GitHub. Busque el ejemplo completo y aprenda a configurar y ejecutar en el Repositorio de ejemplos de código de 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.
```

- Para obtener información sobre la API, consulte CopyObject en la Referencia de la API de AWS SDK para SAP ABAP.

Swift

SDK para Swift

> **Note**
>
> Esto es documentación preliminar para un SDK en versión preliminar. Está sujeta a cambios.

> **Note**
>
> Hay más en GitHub. Busque el ejemplo completo y aprenda a configurar y ejecutar en el Repositorio de ejemplos de código de 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)
    }
```

- Para obtener información sobre la API, consulte CopyObject en la Referencia de la API de AWS SDK para Swift.

Para obtener una lista completa de las guías para desarrolladores del AWS SDK y ejemplos de código, consulte Uso de este servicio con un SDK de AWS. En este tema también se incluye información sobre cómo comenzar a utilizar el SDK y detalles sobre sus versiones anteriores.

Uso de **`CreateBucket`** con un AWS SDK o la CLI

Los siguientes ejemplos de código muestran cómo utilizar `CreateBucket`.

Los ejemplos de acciones son extractos de código de programas más grandes y deben ejecutarse en contexto. Puede ver esta acción en su contexto en los siguientes ejemplos de código:

- Introducción a los buckets y objetos
- Trabajo con objetos con control de versiones

.NET

AWS SDK for .NET

> **Note**
>
> Hay más información en GitHub. Busque el ejemplo completo y aprenda a configurar y ejecutar en el Repositorio de ejemplos de código de 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;
            }
        }
```

Cree un nuevo bucket con el bloqueo de objetos habilitado.

```
   /// <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;
       }
   }
```

- Para obtener información sobre la API, consulte CreateBucket en la Referencia de la API de AWS SDK for .NET.

Bash

AWS CLI con script Bash

> **Note**
>
> Hay más información en GitHub. Busque el ejemplo completo y aprenda a configurar y ejecutar en el Repositorio de ejemplos de código de 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
}
```

- Para obtener información sobre la API, consulte CreateBucket en la Referencia de comandos de AWS CLI.

C++

SDK para C++

> **Note**
>
> Hay más en GitHub. Busque el ejemplo completo y aprenda a configurar y ejecutar en el Repositorio de ejemplos de código de 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();
}
```

- Para obtener información sobre la API, consulte CreateBucket en la Referencia de la API de AWS SDK for C++.

CLI

AWS CLI

Ejemplo 1: Creación de un bucket

En los siguientes ejemplos de `create-bucket` se crea un bucket denominado `my-bucket`:

```
aws s3api create-bucket \
    --bucket my-bucket \
    --region us-east-1
```

Salida:

```
{
    "Location": "/my-bucket"
}
```

Para obtener más información, consulte Crear un bucket en la Guía del usuario de Amazon S3.

Ejemplo 2: Creación de un bucket con propietario obligatorio

En el siguiente ejemplo de `create-bucket` se crea un bucket denominado `my-bucket` que utiliza la configuración Aplicada al propietario del bucket de S3 Object Ownership.

```
aws s3api create-bucket \
    --bucket my-bucket \
    --region us-east-1 \
    --object-ownership BucketOwnerEnforced
```

Salida:

```
{
    "Location": "/my-bucket"
}
```

Para obtener más información, consulte Control de la propiedad de objetos y desactivación de las ACL en la Guía del usuario de Amazon S3.

Ejemplo 3: Creación de un bucket fuera de la región ``us-east-1``

En el siguiente ejemplo `create-bucket`, se crea un bucket denominado `my-bucket` en la región `eu-west-1`. Las regiones situadas fuera de `us-east-1` requieren que se especifique el `LocationConstraint` correspondiente para poder crear el bucket en la región deseada.

```
aws s3api create-bucket \
    --bucket my-bucket \
    --region eu-west-1 \
    --create-bucket-configuration LocationConstraint=eu-west-1
```

Salida:

```
{
    "Location": "http://my-bucket.s3.amazonaws.com/"
}
```

Para obtener más información, consulte Crear un bucket en la Guía del usuario de Amazon S3.

- Para obtener detalles de la API, consulte CreateBucket en la Referencia de comandos de AWS CLI.

Go

SDK para Go V2

> **Note**
>
> Hay más en GitHub. Busque el ejemplo completo y aprenda a configurar y ejecutar en el Repositorio de ejemplos de código de AWS.

Cree un bucket con la configuración predeterminada.

```
// 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
}
```

Cree un bucket con el bloqueo de objetos y espere a que aparezca.

```
// 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
}
```

- Para obtener información sobre la API, consulte CreateBucket en la Referencia de la API de AWS SDK for Go.

Java

SDK para Java 2.x

> **Note**
>
> Hay más en GitHub. Busque el ejemplo completo y aprenda a configurar y ejecutar en el Repositorio de ejemplos de código de AWS.

Crear un bucket.

```
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);
        }
    }
}
```

Cree un nuevo bucket con el bloqueo de objetos habilitado.

```
   // 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");
   }
```

- Para obtener información sobre la API, consulte CreateBucket en la Referencia de la API de AWS SDK for Java 2.x.

JavaScript

SDK para JavaScript (v3)

> **Note**
>
> Hay más información en GitHub. Busque el ejemplo completo y aprenda a configurar y ejecutar en el Repositorio de ejemplos de código de AWS.

Crear el bucket.

```
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);
  }
};
```

- Para obtener información, consulte la Guía para desarrolladores de AWS SDK for JavaScript.
- Para obtener información sobre la API, consulte CreateBucket en la Referencia de la API de AWS SDK for JavaScript.

Kotlin

SDK para Kotlin

> **Note**
>
> Hay más información en GitHub. Busque el ejemplo completo y aprenda a configurar y ejecutar en el Repositorio de ejemplos de código de 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")
    }
}
```

- Para obtener información sobre la API, consulte CreateBucket en la Referencia de la API de AWS SDK para Kotlin.

PHP

SDK para PHP

> **Note**
>
> Hay más en GitHub. Busque el ejemplo completo y aprenda a configurar y ejecutar en el Repositorio de ejemplos de código de AWS.

Crear un bucket.

```
        $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.");
        }
```

- Para obtener información sobre la API, consulte CreateBucket en la Referencia de la API de AWS SDK for PHP.

Python

SDK para Python (Boto3)

> **Note**
>
> Hay más en GitHub. Busque el ejemplo completo y aprenda a configurar y ejecutar en el Repositorio de ejemplos de código de AWS.

Cree un bucket con la configuración predeterminada.

```
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
```

Cree un bucket con control de versiones con una configuración de ciclo de vida.

```
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
```

- Para obtener información sobre la API, consulte CreateBucket en la Referencia de la API de AWS SDK para Python (Boto3).

Ruby

SDK para Ruby

> **Note**
>
> Hay más en GitHub. Busque el ejemplo completo y aprenda a configurar y ejecutar en el Repositorio de ejemplos de código de 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__
```

- Para obtener información sobre la API, consulte CreateBucket en la Referencia de la API de AWS SDK for Ruby.

Rust

SDK para Rust

> **Note**
>
> Hay más información en GitHub. Busque el ejemplo completo y aprenda a configurar y ejecutar en el Repositorio de ejemplos de código de 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
}
```

- Para obtener información sobre la API, consulte CreaCreateBucket en la Referencia de la API de AWS SDK para Rust.

SAP ABAP

SDK de SAP ABAP

> **Note**
>
> Hay más en GitHub. Busque el ejemplo completo y aprenda a configurar y ejecutar en el Repositorio de ejemplos de código de 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.
```

- Para obtener información sobre la API, consulte CreateBucket en la Referencia de la API de AWS SDK para SAP ABAP.

Swift

SDK para Swift

> **Note**
>
> Esto es documentación preliminar para un SDK en versión preliminar. Está sujeta a cambios.

> **Note**
>
> Hay más en GitHub. Busque el ejemplo completo y aprenda a configurar y ejecutar en el Repositorio de ejemplos de código de 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)
```

```
    }
```

- Para obtener información sobre la API, consulte CreateBucket en la Referencia de la API de AWS SDK para Swift.

Para obtener una lista completa de las guías para desarrolladores del AWS SDK y ejemplos de código, consulte Uso de este servicio con un SDK de AWS. En este tema también se incluye información sobre cómo comenzar a utilizar el SDK y detalles sobre sus versiones anteriores.

Uso de **`CreateMultiRegionAccessPoint`** con un AWS SDK o la CLI

En el siguiente ejemplo de código, se muestra cómo usar `CreateMultiRegionAccessPoint`.

Kotlin

SDK para Kotlin

> **Note**
>
> Hay más información en GitHub. Busque el ejemplo completo y aprenda a configurar y ejecutar en el Repositorio de ejemplos de código de AWS.

Configure el cliente de control de S3 para enviar la solicitud a la región 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
        }
```

Cree el punto de acceso de varias regiones.

```
    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"
    }
```

Espere a que el punto de acceso de varias regiones esté 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)
            }
        }
```

- Para obtener información, consulte Guía para desarrolladores del SDK de AWS SDK para Kotlin.
- Para obtener información sobre la API, consulte CreateMultiRegionAccessPoint en la Referencia de la API del SDK de AWS para Kotlin.

Para obtener una lista completa de las guías para desarrolladores del AWS SDK y ejemplos de código, consulte Uso de este servicio con un SDK de AWS. En este tema también se incluye información sobre cómo comenzar a utilizar el SDK y detalles sobre sus versiones anteriores.

Uso de **`CreateMultipartUpload`** con un AWS SDK o la CLI

Los siguientes ejemplos de código muestran cómo utilizar `CreateMultipartUpload`.

Los ejemplos de acciones son extractos de código de programas más grandes y deben ejecutarse en contexto. Puede ver esta acción en su contexto en los siguientes ejemplos de código:

- Ejecución de una copia multiparte
- Ejecución de una carga multiparte
- Usar sumas de comprobación
- Trabajo con la integridad de los objetos de Amazon S3

C++

SDK para C++

> **Note**
>
> Hay más en GitHub. Busque el ejemplo completo y aprenda a configurar y ejecutar en el Repositorio de ejemplos de código de AWS.

```
//! Create a multipart upload.
/*!
    \param bucket: The name of the S3 bucket where the object will be uploaded.
    \param key: The unique identifier (key) for the object within the S3 bucket.
    \param client: The S3 client instance used to perform the upload operation.
    \return Aws::String: Upload ID or empty string if failed.
*/
Aws::String
AwsDoc::S3::createMultipartUpload(const Aws::String &bucket, const Aws::String
 &key,
                                  Aws::S3::Model::ChecksumAlgorithm
 checksumAlgorithm,
                                  const Aws::S3::S3Client &client) {
    Aws::S3::Model::CreateMultipartUploadRequest request;
    request.SetBucket(bucket);
    request.SetKey(key);
```

```
    if (checksumAlgorithm != Aws::S3::Model::ChecksumAlgorithm::NOT_SET) {
        request.SetChecksumAlgorithm(checksumAlgorithm);
    }

    Aws::S3::Model::CreateMultipartUploadOutcome outcome =
            client.CreateMultipartUpload(request);

    Aws::String uploadID;
    if (outcome.IsSuccess()) {
        uploadID = outcome.GetResult().GetUploadId();
    } else {
        std::cerr << "Error creating multipart upload: " <<
 outcome.GetError().GetMessage() << std::endl;
    }

    return uploadID;
}
```

- Para obtener información acerca de la API, consulte CreateMultipartUpload en la Referencia de la API de AWS SDK for C++.

CLI

AWS CLI

El siguiente comando crea una carga multiparte en el bucket `my-bucket` con la clave `multipart/01`:

```
aws s3api create-multipart-upload --bucket my-bucket --key 'multipart/01'
```

Salida:

```
{
    "Bucket": "my-bucket",
    "UploadId":
 "dfRtDYU0WWCCcH43C3WFbkRONycyCpTJJvxu2i5GYkZljF.Yxwh6XG7WfS2vC4to6HiV6Yjlx.cph0gtNBtJ8P3
    "Key": "multipart/01"
}
```

El archivo completo se denominará `01` en una carpeta llamada `multipart` en el bucket `my-bucket`. Guarde el ID de carga, la clave y el nombre del bucket para usarlos con el comando `upload-part`.

- Para obtener información sobre la API, consulte CreateMultipartUpload en la Referencia de comandos de la AWS CLI.

Rust

SDK para Rust

> **Note**
>
> Hay más información en GitHub. Busque el ejemplo completo y aprenda a configurar y ejecutar en el Repositorio de ejemplos de código de AWS.

```
let multipart_upload_res: CreateMultipartUploadOutput = client
    .create_multipart_upload()
    .bucket(&bucket_name)
    .key(&key)
    .send()
    .await
    .unwrap();
```

- Para obtener detalles sobre la API, consulte CreateMultipartUpload en la Referencia de la API del SDK AWS para Rust.

Para obtener una lista completa de las guías para desarrolladores del AWS SDK y ejemplos de código, consulte Uso de este servicio con un SDK de AWS. En este tema también se incluye información sobre cómo comenzar a utilizar el SDK y detalles sobre sus versiones anteriores.

Uso de **`DeleteBucket`** con un AWS SDK o la CLI

Los siguientes ejemplos de código muestran cómo utilizar `DeleteBucket`.

Los ejemplos de acciones son extractos de código de programas más grandes y deben ejecutarse en contexto. Puede ver esta acción en contexto en el siguiente ejemplo de código:

- Introducción a los buckets y objetos

.NET

AWS SDK for .NET

> **Note**
>
> Hay más información en GitHub. Busque el ejemplo completo y aprenda a configurar y ejecutar en el Repositorio de ejemplos de código de 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;
        }
```

- Para obtener información sobre la API, consulte DeleteBucket en la Referencia de la API de AWS SDK for .NET.

Bash

AWS CLI con script Bash

> **Note**
>
> Hay más información en GitHub. Busque el ejemplo completo y aprenda a configurar y ejecutar en el Repositorio de ejemplos de código de 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
}
```

- Para obtener información sobre la API, consulte DeleteBucket en la Referencia de comandos de AWS CLI.

C++

SDK para C++

> **Note**
>
> Hay más en GitHub. Busque el ejemplo completo y aprenda a configurar y ejecutar en el Repositorio de ejemplos de código de 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();
}
```

- Para obtener información sobre la API, consulte DeleteBucket en la Referencia de la API de AWS SDK for C++.

CLI

AWS CLI

El comando siguiente elimina un bucket denominado `my-bucket`:

```
aws s3api delete-bucket --bucket my-bucket --region us-east-1
```

- Para obtener detalles de la API, consulte DeleteBucket en la Referencia de comandos de AWS CLI.

Go

SDK para Go V2

> **Note**
>
> Hay más en GitHub. Busque el ejemplo completo y aprenda a configurar y ejecutar en el Repositorio de ejemplos de código de 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
}
```

- Para obtener información sobre la API, consulte DeleteBucket en la Referencia de la API de AWS SDK for Go.

Java

SDK para Java 2.x

> **Note**
>
> Hay más en GitHub. Busque el ejemplo completo y aprenda a configurar y ejecutar en el Repositorio de ejemplos de código de AWS.

```
        DeleteBucketRequest deleteBucketRequest = DeleteBucketRequest.builder()
                .bucket(bucket)
                .build();

        s3.deleteBucket(deleteBucketRequest);
        s3.close();
```

- Para obtener información sobre la API, consulte DeleteBucket en la Referencia de la API de AWS SDK for Java 2.x.

JavaScript

SDK para JavaScript (v3)

> **Note**
>
> Hay más información en GitHub. Busque el ejemplo completo y aprenda a configurar y ejecutar en el Repositorio de ejemplos de código de AWS.

Elimine el bucket.

```
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);
  }
};
```

- Para obtener información, consulte la Guía para desarrolladores de AWS SDK for JavaScript.
- Para obtener información sobre la API, consulte DeleteBucket en la Referencia de la API de AWS SDK for JavaScript.

PHP

SDK para PHP

> **Note**
>
> Hay más en GitHub. Busque el ejemplo completo y aprenda a configurar y ejecutar en el Repositorio de ejemplos de código de AWS.

Elimine un bucket vacío.

```
        $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.");
        }
```

- Para obtener información sobre la API, consulte DeleteBucket en la Referencia de la API de AWS SDK for PHP.

PowerShell

Herramientas para PowerShell

Ejemplo 1: este comando elimina todos los objetos y las versiones de los objetos del bucket “test-files” y, a continuación, elimina el bucket. El comando solicitará una confirmación antes de continuar. Añada el conmutador -Force para suprimir la confirmación. Tenga en cuenta que los buckets que no estén vacíos no se pueden eliminar.

```
Remove-S3Bucket -BucketName test-files -DeleteBucketContent
```

- Para obtener información sobre la API, consulte DeleteBucket en la Referencia de Cmdlet de AWS Tools for PowerShell.

Python

SDK para Python (Boto3)

> **Note**
>
> Hay más en GitHub. Busque el ejemplo completo y aprenda a configurar y ejecutar en el Repositorio de ejemplos de código de 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
```

- Para obtener información sobre la API, consulte DeleteBucket en la Referencia de la API de AWS SDK para Python (Boto3).

Ruby

SDK para Ruby

> **Note**
>
> Hay más en GitHub. Busque el ejemplo completo y aprenda a configurar y ejecutar en el Repositorio de ejemplos de código de 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
```

- Para obtener información sobre la API, consulte DeleteBucket en la Referencia de la API de AWS SDK for Ruby.

Rust

SDK para Rust

> **Note**
>
> Hay más información en GitHub. Busque el ejemplo completo y aprenda a configurar y ejecutar en el Repositorio de ejemplos de código de 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(())
}
```

- Para obtener información sobre la API, consulte DeleteBucket en la Referencia de la API de AWS SDK para Rust.

SAP ABAP

SDK de SAP ABAP

> **Note**
>
> Hay más en GitHub. Busque el ejemplo completo y aprenda a configurar y ejecutar en el Repositorio de ejemplos de código de 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.
```

- Para obtener información sobre la API, consulte DeleteBucket en la Referencia de la API de AWS SDK para SAP ABAP.

Swift

SDK para Swift

> **Note**
>
> Esto es documentación preliminar para un SDK en versión preliminar. Está sujeta a cambios.

> **Note**
>
> Hay más en GitHub. Busque el ejemplo completo y aprenda a configurar y ejecutar en el Repositorio de ejemplos de código de AWS.

```
public func deleteBucket(name: String) async throws {
    let input = DeleteBucketInput(
        bucket: name
    )
    _ = try await client.deleteBucket(input: input)
}
```

- Para obtener información sobre la API, consulte DeleteBucket en la Referencia de la API de AWS SDK para Swift.

Para obtener una lista completa de las guías para desarrolladores del AWS SDK y ejemplos de código, consulte Uso de este servicio con un SDK de AWS. En este tema también se incluye información sobre cómo comenzar a utilizar el SDK y detalles sobre sus versiones anteriores.

Uso de **DeleteBucketAnalyticsConfiguration** con un AWS SDK o la CLI

Los siguientes ejemplos de código muestran cómo utilizar `DeleteBucketAnalyticsConfiguration`.

CLI

AWS CLI

Eliminar una configuración de análisis de un bucket

En el siguiente ejemplo de `delete-bucket-analytics-configuration`, se elimina la configuración de análisis para el bucket e ID especificados.

```
aws s3api delete-bucket-analytics-configuration \
    --bucket my-bucket \
    --id 1
```

Este comando no genera ninguna salida.

- Para obtener información sobre la API, consulte DeleteBucketAnalyticsConfiguration en la Referencia de comandos de la AWS CLI.

PowerShell

Herramientas para PowerShell

Ejemplo 1: el comando elimina el filtro de análisis con el nombre “testfilter” en el bucket de S3 indicado.

```
Remove-S3BucketAnalyticsConfiguration -BucketName 's3testbucket' -AnalyticsId
 'testfilter'
```

- Para obtener información sobre la API, consulte DeleteBucketAnalyticsConfiguration en la Referencia de Cmdlet de AWS Tools for PowerShell.

Para obtener una lista completa de las guías para desarrolladores del AWS SDK y ejemplos de código, consulte Uso de este servicio con un SDK de AWS. En este tema también se incluye información sobre cómo comenzar a utilizar el SDK y detalles sobre sus versiones anteriores.

Uso de **`DeleteBucketCors`** con un AWS SDK o la CLI

Los siguientes ejemplos de código muestran cómo utilizar `DeleteBucketCors`.

.NET

AWS SDK for .NET

> **Note**
>
> Hay más en GitHub. Busque el ejemplo completo y aprenda a configurar y ejecutar en el Repositorio de ejemplos de código de 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);
        }
```

- Para obtener información sobre la API, consulte DeleteBucketCors en la Referencia de la API de AWS SDK for .NET.

CLI

AWS CLI

El siguiente comando elimina la configuración de uso compartido de recursos entre orígenes desde un bucket denominado `my-bucket`:

```
aws s3api delete-bucket-cors --bucket my-bucket
```

- Para obtener información sobre la API, consulte DeleteBucketCors en la Referencia de comandos de la AWS CLI.

Python

SDK para Python (Boto3)

> **Note**
>
> Hay más en GitHub. Busque el ejemplo completo y aprenda a configurar y ejecutar en el Repositorio de ejemplos de código de 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
```

- Para obtener información sobre la API, consulte DeleteBucketCors en la Referencia de la API de AWS SDK para Python (Boto3).

Ruby

SDK para Ruby

> **Note**
>
> Hay más en GitHub. Busque el ejemplo completo y aprenda a configurar y ejecutar en el Repositorio de ejemplos de código de 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
```

- Para obtener información sobre la API, consulte DeleteBucketCors en la Referencia de la API de AWS SDK for Ruby.

Para obtener una lista completa de las guías para desarrolladores del AWS SDK y ejemplos de código, consulte Uso de este servicio con un SDK de AWS. En este tema también se incluye información sobre cómo comenzar a utilizar el SDK y detalles sobre sus versiones anteriores.

Uso de **`DeleteBucketEncryption`** con un AWS SDK o la CLI

Los siguientes ejemplos de código muestran cómo utilizar `DeleteBucketEncryption`.

CLI

AWS CLI

Eliminar la configuración de cifrado del servidor de un bucket

En el siguiente ejemplo de `delete-bucket-encryption`, se elimina la configuración de cifrado del servidor del bucket especificado.

```
aws s3api delete-bucket-encryption \
    --bucket my-bucket
```

Este comando no genera ninguna salida.

- Para obtener información sobre la API, consulte DeleteBucketEncryption en la Referencia de comandos de la AWS CLI.

PowerShell

Herramientas para PowerShell

Ejemplo 1: esto deshabilita el cifrado habilitado para el bucket de S3 proporcionado.

```
Remove-S3BucketEncryption -BucketName 's3casetestbucket'
```

Salida:

```
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
```

- Para obtener información sobre la API, consulte DeleteBucketEncryption en la Referencia de Cmdlet de AWS Tools for PowerShell.

Para obtener una lista completa de las guías para desarrolladores del AWS SDK y ejemplos de código, consulte Uso de este servicio con un SDK de AWS. En este tema también se incluye información sobre cómo comenzar a utilizar el SDK y detalles sobre sus versiones anteriores.

Uso de **`DeleteBucketInventoryConfiguration`** con un AWS SDK o la CLI

Los siguientes ejemplos de código muestran cómo utilizar `DeleteBucketInventoryConfiguration`.

CLI

AWS CLI

Eliminar la configuración de inventario de un bucket

En el siguiente ejemplo de `delete-bucket-inventory-configuration`, se elimina la configuración de inventario con el ID 1 del bucket especificado.

```
aws s3api delete-bucket-inventory-configuration \
```

```
    --bucket my-bucket \
    --id 1
```

Este comando no genera ninguna salida.

- Para obtener información sobre la API, consulte DeleteBucketInventoryConfiguration en la Referencia de comandos de la AWS CLI.

PowerShell

Herramientas para PowerShell

Ejemplo 1: este comando elimina el inventario denominado “testInventoryName” correspondiente al bucket de S3 en cuestión.

```
Remove-S3BucketInventoryConfiguration -BucketName 's3testbucket' -InventoryId
 'testInventoryName'
```

Salida:

```
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
```

- Para obtener información sobre la API, consulte DeleteBucketInventoryConfiguration en la Referencia de Cmdlet de AWS Tools for PowerShell.

Para obtener una lista completa de las guías para desarrolladores del AWS SDK y ejemplos de código, consulte Uso de este servicio con un SDK de AWS. En este tema también se incluye información sobre cómo comenzar a utilizar el SDK y detalles sobre sus versiones anteriores.

Uso de **`DeleteBucketLifecycle`** con un AWS SDK o la CLI

Los siguientes ejemplos de código muestran cómo utilizar `DeleteBucketLifecycle`.

.NET

AWS SDK for .NET

> **Note**
>
> Hay más en GitHub. Busque el ejemplo completo y aprenda a configurar y ejecutar en el Repositorio de ejemplos de código de 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);
        }
```

- Para obtener información sobre la API, consulte DeleteBucketLifecycle en la Referencia de la API de AWS SDK for .NET.

CLI

AWS CLI

El comando siguiente elimina una configuración del ciclo de vida de un bucket denominado `my-bucket`:

```
aws s3api delete-bucket-lifecycle --bucket my-bucket
```

- Para obtener información sobre la API, consulte DeleteBucketLifecycle en la Referencia de comandos de la AWS CLI.

Python

SDK para Python (Boto3)

> **Note**
>
> Hay más en GitHub. Busque el ejemplo completo y aprenda a configurar y ejecutar en el Repositorio de ejemplos de código de 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
```

- Para obtener información sobre la API, consulte DeleteBucketLifecycle en la Referencia de la API de AWS SDK para Python (Boto3).

Para obtener una lista completa de las guías para desarrolladores del AWS SDK y ejemplos de código, consulte Uso de este servicio con un SDK de AWS. En este tema también se incluye información sobre cómo comenzar a utilizar el SDK y detalles sobre sus versiones anteriores.

Uso de **`DeleteBucketMetricsConfiguration`** con un AWS SDK o la CLI

Los siguientes ejemplos de código muestran cómo utilizar `DeleteBucketMetricsConfiguration`.

CLI

AWS CLI

Eliminar una configuración de métricas de un bucket

En el siguiente ejemplo de `delete-bucket-metrics-configuration`, se elimina la configuración de métricas para el bucket e ID especificados.

```
aws s3api delete-bucket-metrics-configuration \
    --bucket my-bucket \
    --id 123
```

Este comando no genera ninguna salida.

- Para obtener información sobre la API, consulte DeleteBucketMetricsConfiguration en la Referencia de comandos de la AWS CLI.

PowerShell

Herramientas para PowerShell

Ejemplo 1: el comando elimina el filtro de métricas con el nombre “testmetrics” en el bucket de S3 indicado.

```
Remove-S3BucketMetricsConfiguration -BucketName 's3testbucket' -MetricsId
 'testmetrics'
```

- Para obtener información sobre la API, consulte DeleteBucketMetricsConfiguration en la Referencia de Cmdlet de AWS Tools for PowerShell.

Para obtener una lista completa de las guías para desarrolladores del AWS SDK y ejemplos de código, consulte Uso de este servicio con un SDK de AWS. En este tema también se incluye información sobre cómo comenzar a utilizar el SDK y detalles sobre sus versiones anteriores.

Uso de **`DeleteBucketPolicy`** con un AWS SDK o la CLI

Los siguientes ejemplos de código muestran cómo utilizar `DeleteBucketPolicy`.

C++

SDK para C++

> **Note**
>
> Hay más en GitHub. Busque el ejemplo completo y aprenda a configurar y ejecutar en el Repositorio de ejemplos de código de 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();
}
```

- Para obtener información sobre la API, consulte DeleteBucketPolicy en la Referencia de la API de AWS SDK for C++.

CLI

AWS CLI

El comando siguiente elimina una política de bucket de un bucket denominado `my-bucket`:

```
aws s3api delete-bucket-policy --bucket my-bucket
```

- Para obtener información sobre la API, consulte DeleteBucketPolicy en la Referencia de comandos de la AWS CLI.

Java

SDK para Java 2.x

> **Note**
>
> Hay más en GitHub. Busque el ejemplo completo y aprenda a configurar y ejecutar en el Repositorio de ejemplos de código de 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);
        }
    }
}
```

- Para obtener información sobre la API, consulte DeleteBucketPolicy en la Referencia de la API de AWS SDK for Java 2.x.

JavaScript

SDK para JavaScript (v3)

> **Note**
>
> Hay más información en GitHub. Busque el ejemplo completo y aprenda a configurar y ejecutar en el Repositorio de ejemplos de código de AWS.

Elimine la política del bucket.

```
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);
  }
};
```

- Para obtener información, consulte la Guía para desarrolladores de AWS SDK for JavaScript.
- Para obtener información sobre la API, consulte DeleteBucketPolicy en la Referencia de la API de AWS SDK for JavaScript.

Kotlin

SDK para Kotlin

> **Note**
>
> Hay más información en GitHub. Busque el ejemplo completo y aprenda a configurar y ejecutar en el Repositorio de ejemplos de código de AWS.

```
suspend fun deleteS3BucketPolicy(bucketName: String?) {
    val request =
        DeleteBucketPolicyRequest {
            bucket = bucketName
        }

    S3Client { region = "us-east-1" }.use { s3 ->
        s3.deleteBucketPolicy(request)
        println("Done!")
    }
}
```

- Para obtener información sobre la API, consulte DeleteBucketPolicy en la Referencia de la API de AWS SDK para Kotlin.

PowerShell

Herramientas para PowerShell

Ejemplo 1: el comando elimina la política de bucket asociada al bucket de S3 indicado.

```
Remove-S3BucketPolicy -BucketName 's3testbucket'
```

- Para obtener información sobre la API, consulte DeleteBucketPolicy en la Referencia de Cmdlet de AWS Tools for PowerShell.

Python

SDK para Python (Boto3)

> **Note**
>
> Hay más en GitHub. Busque el ejemplo completo y aprenda a configurar y ejecutar en el Repositorio de ejemplos de código de 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
```

- Para obtener información sobre la API, consulte DeleteBucketPolicy en la Referencia de la API de AWS SDK para Python (Boto3).

Ruby

SDK para Ruby

> **Note**
>
> Hay más en GitHub. Busque el ejemplo completo y aprenda a configurar y ejecutar en el Repositorio de ejemplos de código de 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
```

- Para obtener información sobre la API, consulte DeleteBucketPolicy en la Referencia de la API de AWS SDK for Ruby.

Para obtener una lista completa de las guías para desarrolladores del AWS SDK y ejemplos de código, consulte Uso de este servicio con un SDK de AWS. En este tema también se incluye información sobre cómo comenzar a utilizar el SDK y detalles sobre sus versiones anteriores.

Uso de **`DeleteBucketReplication`** con un AWS SDK o la CLI

Los siguientes ejemplos de código muestran cómo utilizar `DeleteBucketReplication`.

CLI

AWS CLI

El siguiente comando elimina la configuración de replicación de un bucket denominado `my-bucket`:

```
aws s3api delete-bucket-replication --bucket my-bucket
```

- Para obtener información sobre la API, consulte DeleteBucketReplication en la Referencia de comandos de la AWS CLI.

PowerShell

Herramientas para PowerShell

Ejemplo 1: elimina la configuración de replicación asociada al bucket denominado “mybucket”. Tenga en cuenta que esta operación necesita permiso para la acción s3:DeleteReplicationConfiguration. Se le solicitará la confirmación antes de continuar con la operación; para suprimir la confirmación, utilice el conmutador -Force.

```
Remove-S3BucketReplication -BucketName mybucket
```

- Para obtener información sobre la API, consulte DeleteBucketReplication en la Referencia de Cmdlet de AWS Tools for PowerShell.

Para obtener una lista completa de las guías para desarrolladores del AWS SDK y ejemplos de código, consulte Uso de este servicio con un SDK de AWS. En este tema también se incluye información sobre cómo comenzar a utilizar el SDK y detalles sobre sus versiones anteriores.

Uso de **`DeleteBucketTagging`** con un AWS SDK o la CLI

Los siguientes ejemplos de código muestran cómo utilizar `DeleteBucketTagging`.

CLI

AWS CLI

El siguiente comando elimina la configuración de etiquetado de un bucket denominado `my-bucket`:

```
aws s3api delete-bucket-tagging --bucket my-bucket
```

- Para obtener información sobre la API, consulte DeleteBucketTagging en la Referencia de comandos de la AWS CLI.

PowerShell

Herramientas para PowerShell

Ejemplo 1: este comando elimina todas las etiquetas asociadas al bucket de S3 indicado.

```
Remove-S3BucketTagging -BucketName 's3testbucket'
```

Salida:

```
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
```

- Para obtener información sobre la API, consulte DeleteBucketTagging en la Referencia de Cmdlet de AWS Tools for PowerShell.

Para obtener una lista completa de las guías para desarrolladores del AWS SDK y ejemplos de código, consulte Uso de este servicio con un SDK de AWS. En este tema también se incluye información sobre cómo comenzar a utilizar el SDK y detalles sobre sus versiones anteriores.

Uso de **`DeleteBucketWebsite`** con un AWS SDK o la CLI

Los siguientes ejemplos de código muestran cómo utilizar `DeleteBucketWebsite`.

C++

SDK para C++

> **Note**
>
> Hay más en GitHub. Busque el ejemplo completo y aprenda a configurar y ejecutar en el Repositorio de ejemplos de código de 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();
}
```

- Para obtener información sobre la API, consulte DeleteBucketWebsite en la Referencia de la API de AWS SDK for C++.

CLI

AWS CLI

El siguiente comando elimina la configuración de un sitio web de un bucket denominado `my-bucket`:

```
aws s3api delete-bucket-website --bucket my-bucket
```

- Para obtener información sobre la API, consulte DeleteBucketWebsite en la Referencia de comandos de la AWS CLI.

Java

SDK para Java 2.x

> **Note**
>
> Hay más en GitHub. Busque el ejemplo completo y aprenda a configurar y ejecutar en el Repositorio de ejemplos de código de 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);
        }
    }
```

```
}
```

- Para obtener información sobre la API, consulte DeleteBucketWebsite en la Referencia de la API de AWS SDK for Java 2.x.

JavaScript

SDK para JavaScript (v3)

> **Note**
>
> Hay más información en GitHub. Busque el ejemplo completo y aprenda a configurar y ejecutar en el Repositorio de ejemplos de código de AWS.

Elimine la configuración del sitio web del bucket.

```
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);
  }
};
```

- Para obtener información, consulte la Guía para desarrolladores de AWS SDK for JavaScript.

- Para obtener información sobre la API, consulte DeleteBucketWebsite en la Referencia de la API de AWS SDK for JavaScript.

PowerShell

Herramientas para PowerShell

Ejemplo 1: este comando deshabilita la propiedad de alojamiento de sitios web estáticos del bucket de S3 indicado.

```
Remove-S3BucketWebsite -BucketName 's3testbucket'
```

Salida:

```
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
```

- Para obtener información sobre la API, consulte DeleteBucketWebsite en la Referencia de Cmdlet de AWS Tools for PowerShell.

Para obtener una lista completa de las guías para desarrolladores del AWS SDK y ejemplos de código, consulte Uso de este servicio con un SDK de AWS. En este tema también se incluye información sobre cómo comenzar a utilizar el SDK y detalles sobre sus versiones anteriores.

Uso de **`DeleteObject`** con un AWS SDK o la CLI

Los siguientes ejemplos de código muestran cómo utilizar `DeleteObject`.

Los ejemplos de acciones son extractos de código de programas más grandes y deben ejecutarse en contexto. Puede ver esta acción en su contexto en los siguientes ejemplos de código:

- Trabajo con la integridad de los objetos de Amazon S3
- Trabajo con objetos con control de versiones

.NET

AWS SDK for .NET

> **Note**
>
> Hay más información en GitHub. Busque el ejemplo completo y aprenda a configurar y ejecutar en el Repositorio de ejemplos de código de AWS.

Elimine un objeto de un bucket de S3 no versionado.

```
    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.");
            }
        }
    }
```

Elimine un objeto de un bucket de S3 versionado.

```
    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;
        }
    }
```

- Para obtener información sobre la API, consulte DeleteObject en la Referencia de la API de AWS SDK for .NET.

Bash

AWS CLI con script Bash

> **Note**
>
> Hay más información en GitHub. Busque el ejemplo completo y aprenda a configurar y ejecutar en el Repositorio de ejemplos de código de 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
}
```

- Para obtener información sobre la API, consulte DeleteObject en la Referencia de comandos de AWS CLI.

C++

SDK para C++

> **Note**
>
> Hay más en GitHub. Busque el ejemplo completo y aprenda a configurar y ejecutar en el Repositorio de ejemplos de código de 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();
}
```

- Para obtener información sobre la API, consulte DeleteObject en la Referencia de la API de AWS SDK for C++.

CLI

AWS CLI

El comando siguiente elimina un objeto denominado `test.txt` de un bucket denominado `my-bucket`:

```
aws s3api delete-object --bucket my-bucket --key test.txt
```

Si el control de versiones del bucket está activado, el resultado contendrá el ID de versión del marcador de eliminación:

```
{
  "VersionId": "9_gKg5vG56F.TTEUdwkxGpJ3tNDlWlGq",
  "DeleteMarker": true
}
```

Para obtener más información acerca de la eliminación de objetos, consulte Eliminación de objetos en la Guía para desarrolladores de Amazon S3.

- Para obtener información sobre la API, consulte DeleteObject en la Referencia de comandos de AWS CLI.

Go

SDK para Go V2

> **Note**
>
> Hay más en GitHub. Busque el ejemplo completo y aprenda a configurar y ejecutar en el Repositorio de ejemplos de código de 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
}
```

- Para obtener información sobre la API, consulte DeleteObject en la Referencia de la API de AWS SDK for Go.

JavaScript

SDK para JavaScript (v3)

> **Note**
>
> Hay más información en GitHub. Busque el ejemplo completo y aprenda a configurar y ejecutar en el Repositorio de ejemplos de código de AWS.

Elimine un objeto.

```
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);
  }
};
```

- Para obtener información sobre la API, consulte DeleteObject en la Referencia de la API de AWS SDK for JavaScript.

Python

SDK para Python (Boto3)

> **Note**
>
> Hay más en GitHub. Busque el ejemplo completo y aprenda a configurar y ejecutar en el Repositorio de ejemplos de código de AWS.

Elimine un objeto.

```
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
```

Revierta un objeto a una versión anterior y elimine versiones posteriores del objeto.

```
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}."
        )
```

Reactive un objeto eliminado quitando el marcador de eliminación activo del objeto.

```
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)
```

Cree un controlador Lambda que elimine un marcador de eliminación de un objeto de S3. Este controlador se puede utilizar para limpiar de forma eficiente marcadores de eliminación extraños en un bucket con control de versiones.

```
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,
    }
```

- Para obtener información sobre la API, consulte DeleteObject en la Referencia de la API de AWS SDK para Python (Boto3).

Rust

SDK para Rust

> **Note**
>
> Hay más información en GitHub. Busque el ejemplo completo y aprenda a configurar y ejecutar en el Repositorio de ejemplos de código de 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(())
}
```

- Para obtener información sobre la API, consulte DeleteObject en la Referencia de la API de AWS SDK para Rust.

SAP ABAP

SDK de SAP ABAP

> **Note**
>
> Hay más en GitHub. Busque el ejemplo completo y aprenda a configurar y ejecutar en el Repositorio de ejemplos de código de 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.
```

- Para obtener información sobre la API, consulte DeleteObject en la Referencia de la API de AWS SDK para SAP ABAP.

Swift

SDK para Swift

> **Note**
>
> Esto es documentación preliminar para un SDK en versión preliminar. Está sujeta a cambios.

> **Note**
>
> Hay más en GitHub. Busque el ejemplo completo y aprenda a configurar y ejecutar en el Repositorio de ejemplos de código de 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
    }
}
```

- Para obtener información sobre la API, consulte DeleteObject en la Referencia de la API de AWS SDK para Swift.

Para obtener una lista completa de las guías para desarrolladores del AWS SDK y ejemplos de código, consulte Uso de este servicio con un SDK de AWS. En este tema también se incluye información sobre cómo comenzar a utilizar el SDK y detalles sobre sus versiones anteriores.

Uso de **`DeleteObjectTagging`** con un AWS SDK o la CLI

Los siguientes ejemplos de código muestran cómo utilizar `DeleteObjectTagging`.

CLI

AWS CLI

Eliminar los conjuntos de etiquetas de un objeto

En el siguiente ejemplo de `delete-object-tagging`, se elimina del objeto `doc1.rtf` la etiqueta con la clave especificada.

```
aws s3api delete-object-tagging \
    --bucket my-bucket \
    --key doc1.rtf
```

Este comando no genera ninguna salida.

- Para obtener información sobre la API, consulte DeleteObjectTagging en la Referencia de comandos de la AWS CLI.

PowerShell

Herramientas para PowerShell

Ejemplo 1: este comando elimina todas las etiquetas asociadas con el objeto con la clave “testfile.txt” en el bucket de S3 indicado.

```
Remove-S3ObjectTagSet -Key 'testfile.txt' -BucketName 's3testbucket' -Select
 '^Key'
```

Salida:

```
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
```

- Para obtener información sobre la API, consulte DeleteObjectTagging en la Referencia de Cmdlet de AWS Tools for PowerShell.

Para obtener una lista completa de las guías para desarrolladores del AWS SDK y ejemplos de código, consulte Uso de este servicio con un SDK de AWS. En este tema también se incluye información sobre cómo comenzar a utilizar el SDK y detalles sobre sus versiones anteriores.

Uso de **`DeleteObjects`** con un AWS SDK o la CLI

Los siguientes ejemplos de código muestran cómo utilizar `DeleteObjects`.

Los ejemplos de acciones son extractos de código de programas más grandes y deben ejecutarse en contexto. Puede ver esta acción en contexto en el siguiente ejemplo de código:

- Introducción a los buckets y objetos

.NET

AWS SDK for .NET

> **Note**
>
> Hay más información en GitHub. Busque el ejemplo completo y aprenda a configurar y ejecutar en el Repositorio de ejemplos de código de AWS.

Elimine todos los objetos de un bucket de 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;
            }
        }
```

Elimine varios objetos de un bucket de S3 no versionado.

```
    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;
        }
    }
```

Elimine varios objetos de un bucket de S3 versionado.

```
    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;
        }
    }
```

- Para obtener información sobre la API, consulte DeleteObjects en la Referencia de la API de AWS SDK for .NET.

Bash

AWS CLI con script Bash

> **Note**
>
> Hay más información en GitHub. Busque el ejemplo completo y aprenda a configurar y ejecutar en el Repositorio de ejemplos de código de 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
}
```

- Para obtener información sobre la API, consulte DeleteObjects en la Referencia de comandos de AWS CLI.

C++

SDK para C++

> **Note**
>
> Hay más en GitHub. Busque el ejemplo completo y aprenda a configurar y ejecutar en el Repositorio de ejemplos de código de 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();
}
```

- Para obtener información sobre la API, consulte DeleteObjects en la Referencia de la API de AWS SDK for C++.

CLI

AWS CLI

El comando siguiente elimina un objeto denominado de un bucket denominado `my-bucket`:

```
aws s3api delete-objects --bucket my-bucket --delete file://delete.json
```

`delete.json` es un documento JSON en el directorio actual que especifica el objeto que se va a eliminar:

```
{
  "Objects": [
    {
      "Key": "test1.txt"
    }
  ],
  "Quiet": false
}
```

Salida:

```
{
    "Deleted": [
        {
            "DeleteMarkerVersionId": "mYAT5Mc6F7aeUL8SS7FAAqUPO1koHwzU",
            "Key": "test1.txt",
            "DeleteMarker": true
        }
    ]
}
```

- Para obtener detalles de la API, consulte DeleteObjects en la Referencia de comandos de AWS CLI.

Go

SDK para Go V2

> **Note**
>
> Hay más en GitHub. Busque el ejemplo completo y aprenda a configurar y ejecutar en el Repositorio de ejemplos de código de 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
}
```

- Para obtener información sobre la API, consulte DeleteObjects en la Referencia de la API de AWS SDK for Go.

Java

SDK para Java 2.x

> **Note**
>
> Hay más en GitHub. Busque el ejemplo completo y aprenda a configurar y ejecutar en el Repositorio de ejemplos de código de 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);
        }
    }
}
```

- Para obtener información sobre la API, consulte DeleteObjects en la Referencia de la API de AWS SDK for Java 2.x.

JavaScript

SDK para JavaScript (v3)

> **Note**
>
> Hay más información en GitHub. Busque el ejemplo completo y aprenda a configurar y ejecutar en el Repositorio de ejemplos de código de AWS.

Elimine varios objetos.

```
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);
  }
};
```

- Para obtener información sobre la API, consulte DeleteObjects en la Referencia de la API de AWS SDK for JavaScript.

Kotlin

SDK para Kotlin

> **Note**
>
> Hay más información en GitHub. Busque el ejemplo completo y aprenda a configurar y ejecutar en el Repositorio de ejemplos de código de 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")
    }
}
```

- Para obtener información sobre la API, consulte DeleteObjects en la Referencia de la API de AWS SDK para Kotlin.

PHP

SDK para PHP

> **Note**
>
> Hay más en GitHub. Busque el ejemplo completo y aprenda a configurar y ejecutar en el Repositorio de ejemplos de código de AWS.

Elimine un conjunto de objetos de una lista de claves.

```
        $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.");
        }
```

- Para obtener información sobre la API, consulte DeleteObjects en la Referencia de la API de AWS SDK for PHP.

PowerShell

Herramientas para PowerShell

Ejemplo 1: este comando elimina el objeto “sample.txt” del bucket “test-files”. Antes de ejecutar el comando, se le solicitará que lo confirme; para suprimir el mensaje, utilice el conmutador -Force.

```
Remove-S3Object -BucketName test-files -Key sample.txt
```

Ejemplo 2: este comando elimina la versión especificada del objeto “sample.txt” del bucket “test-files”, suponiendo que el bucket se haya configurado para habilitar las versiones de los objetos.

```
Remove-S3Object -BucketName test-files -Key sample.txt -VersionId
 HLbxnx6V9omT6AQYVpks8mmFKQcejpqt
```

Ejemplo 3: este comando elimina los objetos “sample1.txt”, “sample2.txt” y “sample3.txt” del bucket “test-files” como una sola operación por lotes. La respuesta del servicio mostrará una lista de todas las claves procesadas, independientemente del estado de éxito o error de la eliminación. Para obtener únicamente los errores de las claves que el servicio no ha podido procesar, añada el parámetro -ReportErrorsOnly (este parámetro también se puede especificar con el alias -Quiet).

```
Remove-S3Object -BucketName test-files -KeyCollection @( "sample1.txt",
 "sample2.txt", "sample3.txt" )
```

Ejemplo 4: este ejemplo utiliza una expresión en línea con el parámetro -KeyCollection para obtener las claves de los objetos que se van a eliminar. Get-S3Object devuelve una colección de instancias de Amazon.S3.Model.S3Object, cada una de las cuales tiene un elemento clave de tipo cadena que identifica el objeto.

```
Remove-S3Object -bucketname "test-files" -KeyCollection (Get-S3Object "test-
files" -KeyPrefix "prefix/subprefix" | select -ExpandProperty Key)
```

Ejemplo 5: este ejemplo obtiene todos los objetos que tienen un prefijo de clave “prefijo/subprefijo” en el bucket y los elimina. Tenga en cuenta que los objetos entrantes se procesan de uno en uno. En el caso de colecciones grandes, plantéese la posibilidad de pasarlas al parámetro -InputObject (alias -S3ObjectCollection) del cmdlet para permitir que la eliminación se realice por lotes con una sola llamada al servicio.

```
Get-S3Object -BucketName "test-files" -KeyPrefix "prefix/subprefix" | Remove-
S3Object -Force
```

Ejemplo 6: en este ejemplo, se canaliza al cmdlet para su eliminación una colección de instancias de Amazon.S3.Model.S3ObjectVersion que representan marcadores de eliminación. Tenga en cuenta que los objetos entrantes se procesan de uno en uno. En el caso de colecciones grandes, plantéese la posibilidad de pasarlas al parámetro -InputObject (alias -S3ObjectCollection) del cmdlet para permitir que la eliminación se realice por lotes con una sola llamada al servicio.

```
(Get-S3Version -BucketName "test-files").Versions | Where {$_.IsDeleteMarker -eq
 "True"} | Remove-S3Object -Force
```

Ejemplo 7: este script muestra cómo eliminar por lotes un conjunto de objetos (en este caso, marcadores de eliminación) mediante la creación de una matriz de objetos para utilizarlos con el parámetro -KeyAndVersionCollection.

```
$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
```

- Para obtener información sobre la API, consulte DeleteObjects en la Referencia de Cmdlet de AWS Tools for PowerShell.

Python

SDK para Python (Boto3)

> **Note**
>
> Hay más en GitHub. Busque el ejemplo completo y aprenda a configurar y ejecutar en el Repositorio de ejemplos de código de AWS.

Elimine un conjunto de objetos mediante una lista de claves de objeto.

```
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
```

Elimine todos los objetos de un bucket.

```
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
```

Elimine de forma permanente un objeto con control de versiones mediante la eliminación de todas sus versiones.

```
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
```

- Para obtener información sobre la API, consulte DeleteObjects en la Referencia de la API de AWS SDK para Python (Boto3).

Ruby

SDK para Ruby

> **Note**
>
> Hay más en GitHub. Busque el ejemplo completo y aprenda a configurar y ejecutar en el Repositorio de ejemplos de código de 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
```

- Para obtener información sobre la API, consulte DeleteObjects en la Referencia de la API de AWS SDK for Ruby.

Rust

SDK para Rust

> **Note**
>
> Hay más información en GitHub. Busque el ejemplo completo y aprenda a configurar y ejecutar en el Repositorio de ejemplos de código de 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.",
        )),
    }
}
```

- Para obtener información sobre la API, consulte DeleteObjects en la Referencia de la API de AWS SDK para Rust.

Swift

SDK para Swift

> **Note**
>
> Esto es documentación preliminar para un SDK en versión preliminar. Está sujeta a cambios.

> **Note**
>
> Hay más en GitHub. Busque el ejemplo completo y aprenda a configurar y ejecutar en el Repositorio de ejemplos de código de 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
        }
    }
```

- Para obtener información sobre la API, consulte DeleteObjects en la Referencia de la API de AWS SDK para Swift.

Para obtener una lista completa de las guías para desarrolladores del AWS SDK y ejemplos de código, consulte Uso de este servicio con un SDK de AWS. En este tema también se incluye información sobre cómo comenzar a utilizar el SDK y detalles sobre sus versiones anteriores.

Uso de **`DeletePublicAccessBlock`** con un AWS SDK o la CLI

Los siguientes ejemplos de código muestran cómo utilizar `DeletePublicAccessBlock`.

CLI

AWS CLI

Eliminar la configuración de bloqueo de acceso público de un bucket

En el siguiente ejemplo de `delete-public-access-block`, se elimina la configuración de bloqueo de acceso público en el bucket especificado.

```
aws s3api delete-public-access-block \
    --bucket my-bucket
```

Este comando no genera ninguna salida.

- Para obtener información sobre la API, consulte DeletePublicAccessBlock en la Referencia de comandos de la AWS CLI.

PowerShell

Herramientas para PowerShell

Ejemplo 1: este comando desactiva la configuración de bloqueo de acceso público para el bucket en cuestión.

```
Remove-S3PublicAccessBlock -BucketName 's3testbucket' -Force -Select
 '^BucketName'
```

Salida:

```
s3testbucket
```

- Para obtener información sobre la API, consulte DeletePublicAccessBlock en la Referencia de Cmdlet de AWS Tools for PowerShell.

Para obtener una lista completa de las guías para desarrolladores del AWS SDK y ejemplos de código, consulte Uso de este servicio con un SDK de AWS. En este tema también se incluye información sobre cómo comenzar a utilizar el SDK y detalles sobre sus versiones anteriores.

Uso de **`GetBucketAccelerateConfiguration`** con un AWS SDK o la CLI

Los siguientes ejemplos de código muestran cómo utilizar `GetBucketAccelerateConfiguration`.

CLI

AWS CLI

Recuperar la configuración acelerada de un bucket

En el siguiente ejemplo de `get-bucket-accelerate-configuration`, se recupera la configuración acelerada para el bucket especificado.

```
aws s3api get-bucket-accelerate-configuration \
    --bucket my-bucket
```

Salida:

```
{
    "Status": "Enabled"
}
```

- Para obtener información sobre la API, consulte GetBucketAccelerateConfiguration en la Referencia de comandos de la AWS CLI.

PowerShell

Herramientas para PowerShell

Ejemplo 1: este comando devuelve el valor Enabled si la configuración de aceleración de transferencia está habilitada para el bucket especificado.

```
Get-S3BucketAccelerateConfiguration -BucketName 's3testbucket'
```

Salida:

```
Value
-----
Enabled
```

- Para obtener información sobre la API, consulte GetBucketAccelerateConfiguration en la Referencia de Cmdlet de AWS Tools for PowerShell.

Para obtener una lista completa de las guías para desarrolladores del AWS SDK y ejemplos de código, consulte Uso de este servicio con un SDK de AWS. En este tema también se incluye información sobre cómo comenzar a utilizar el SDK y detalles sobre sus versiones anteriores.

Uso de **`GetBucketAcl`** con un AWS SDK o la CLI

Los siguientes ejemplos de código muestran cómo utilizar `GetBucketAcl`.

Los ejemplos de acciones son extractos de código de programas más grandes y deben ejecutarse en contexto. Puede ver esta acción en contexto en el siguiente ejemplo de código:

- Administrar listas de control de acceso (ACL)

.NET

AWS SDK for .NET

> **Note**
>
> Hay más información en GitHub. Busque el ejemplo completo y aprenda a configurar y ejecutar en el Repositorio de ejemplos de código de 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;
        }
```

- Para obtener información sobre la API, consulte GetBucketAcl en la Referencia de la API de AWS SDK for .NET.

C++

SDK para C++

> **Note**
>
> Hay más en GitHub. Busque el ejemplo completo y aprenda a configurar y ejecutar en el Repositorio de ejemplos de código de 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";
}
```

- Para obtener información sobre la API, consulte GetBucketAcl en la Referencia de la API de AWS SDK for C++.

CLI

AWS CLI

El siguiente comando recupera la lista de control de acceso de un bucket denominado `my-bucket`:

```
aws s3api get-bucket-acl --bucket my-bucket
```

Salida:

```
{
    "Owner": {
        "DisplayName": "my-username",
        "ID": "7009a8971cd538e11f6b6606438875e7c86c5b672f46db45460ddcd087d36c32"
    },
    "Grants": [
        {
            "Grantee": {
                "DisplayName": "my-username",
                "ID":
 "7009a8971cd538e11f6b6606438875e7c86c5b672f46db45460ddcd087d36c32"
            },
            "Permission": "FULL_CONTROL"
        }
    ]
}
```

- Para obtener información sobre la API, consulte GetBucketAcl en la Referencia de comandos de la AWS CLI.

Java

SDK para Java 2.x

> **Note**
>
> Hay más en GitHub. Busque el ejemplo completo y aprenda a configurar y ejecutar en el Repositorio de ejemplos de código de 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 "";
    }
}
```

- Para obtener información sobre la API, consulte GetBucketAcl en la Referencia de la API de AWS SDK for Java 2.x.

JavaScript

SDK para JavaScript (v3)

> **Note**
>
> Hay más información en GitHub. Busque el ejemplo completo y aprenda a configurar y ejecutar en el Repositorio de ejemplos de código de AWS.

Obtenga los permisos de 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);
  }
};
```

- Para obtener información, consulte la Guía para desarrolladores de AWS SDK for JavaScript.
- Para obtener información sobre la API, consulte GetBucketAcl en la Referencia de la API de AWS SDK for JavaScript.

Python

SDK para Python (Boto3)

> **Note**
>
> Hay más en GitHub. Busque el ejemplo completo y aprenda a configurar y ejecutar en el Repositorio de ejemplos de código de 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
```

- Para obtener información sobre la API, consulte GetBucketAcl en la Referencia de la API de AWS SDK para Python (Boto3).

Para obtener una lista completa de las guías para desarrolladores del AWS SDK y ejemplos de código, consulte Uso de este servicio con un SDK de AWS. En este tema también se incluye información sobre cómo comenzar a utilizar el SDK y detalles sobre sus versiones anteriores.

Uso de **GetBucketAnalyticsConfiguration** con un AWS SDK o la CLI

Los siguientes ejemplos de código muestran cómo utilizar `GetBucketAnalyticsConfiguration`.

CLI

AWS CLI

Recuperar la configuración de análisis de un bucket con un ID específico

En el siguiente ejemplo de `get-bucket-analytics-configuration`, se muestra la configuración de análisis para el bucket e ID especificados.

```
aws s3api get-bucket-analytics-configuration \
    --bucket my-bucket \
    --id 1
```

Salida:

```
{
    "AnalyticsConfiguration": {
        "StorageClassAnalysis": {},
        "Id": "1"
    }
}
```

- Para obtener información sobre la API, consulte GetBucketAnalyticsConfiguration en la Referencia de comandos de la AWS CLI.

PowerShell

Herramientas para PowerShell

Ejemplo 1: este comando devuelve los detalles del filtro de análisis con el nombre “testfilter” en el bucket de S3 indicado.

```
Get-S3BucketAnalyticsConfiguration -BucketName 's3testbucket' -AnalyticsId
 'testfilter'
```

- Para obtener información sobre la API, consulte GetBucketAnalyticsConfiguration en la Referencia de Cmdlet de AWS Tools for PowerShell.

Para obtener una lista completa de las guías para desarrolladores del AWS SDK y ejemplos de código, consulte Uso de este servicio con un SDK de AWS. En este tema también se incluye información sobre cómo comenzar a utilizar el SDK y detalles sobre sus versiones anteriores.

Uso de **`GetBucketCors`** con un AWS SDK o la CLI

Los siguientes ejemplos de código muestran cómo utilizar `GetBucketCors`.

.NET

AWS SDK for .NET

> **Note**
>
> Hay más en GitHub. Busque el ejemplo completo y aprenda a configurar y ejecutar en el Repositorio de ejemplos de código de 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;
        }
```

- Para ver la información de la API, consulte GetBucketCors en la Referencia de la API de AWS SDK for .NET.

CLI

AWS CLI

El siguiente comando recupera la configuración de uso compartido de recursos entre orígenes para un bucket denominado `my-bucket`:

```
aws s3api get-bucket-cors --bucket my-bucket
```

Salida:

```
{
    "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": [
                "*"
            ]
        }
    ]
}
```

- Para obtener información sobre la API, consulte GetBucketCors en la Referencia de comandos de la AWS CLI.

JavaScript

SDK para JavaScript (v3)

> **Note**
>
> Hay más información en GitHub. Busque el ejemplo completo y aprenda a configurar y ejecutar en el Repositorio de ejemplos de código de AWS.

Obtenga la política de CORS para el bucket.

```
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);
  }
};
```

- Para obtener información, consulte la Guía para desarrolladores de AWS SDK for JavaScript.
- Para ver la información de la API, consulte GetBucketCors en la Referencia de la API de AWS SDK for JavaScript.

Python

SDK para Python (Boto3)

> **Note**
>
> Hay más en GitHub. Busque el ejemplo completo y aprenda a configurar y ejecutar en el Repositorio de ejemplos de código de 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
```

- Para obtener información sobre las API, consulte GetBucketCors en la Referencia de la API de AWS SDK para Python (Boto3).

Ruby

SDK para Ruby

> **Note**
>
> Hay más en GitHub. Busque el ejemplo completo y aprenda a configurar y ejecutar en el Repositorio de ejemplos de código de 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
```

- Para ver la información de la API, consulte GetBucketCors en la Referencia de la API de AWS SDK for Ruby.

Para obtener una lista completa de las guías para desarrolladores del AWS SDK y ejemplos de código, consulte Uso de este servicio con un SDK de AWS. En este tema también se incluye información sobre cómo comenzar a utilizar el SDK y detalles sobre sus versiones anteriores.

Uso de **`GetBucketEncryption`** con un AWS SDK o la CLI

Los siguientes ejemplos de código muestran cómo utilizar `GetBucketEncryption`.

CLI

AWS CLI

Recuperar la configuración de cifrado del servidor de un bucket

En el siguiente ejemplo de `get-bucket-encryption`, se recupera la configuración de cifrado del lado del servidor del bucket `my-bucket`.

```
aws s3api get-bucket-encryption \
    --bucket my-bucket
```

Salida:

```
{
    "ServerSideEncryptionConfiguration": {
        "Rules": [
            {
                "ApplyServerSideEncryptionByDefault": {
                    "SSEAlgorithm": "AES256"
                }
            }
        ]
    }
}
```

- Para obtener información sobre la API, consulte GetBucketEncryption en la Referencia de comandos de la AWS CLI.

PowerShell

Herramientas para PowerShell

Ejemplo 1: este comando devuelve todas las reglas de cifrado del servidor asociadas al bucket determinado.

```
Get-S3BucketEncryption -BucketName 's3casetestbucket'
```

- Para obtener información sobre la API, consulte GetBucketEncryption en la Referencia de Cmdlet de AWS Tools for PowerShell.

Para obtener una lista completa de las guías para desarrolladores del AWS SDK y ejemplos de código, consulte Uso de este servicio con un SDK de AWS. En este tema también se incluye información sobre cómo comenzar a utilizar el SDK y detalles sobre sus versiones anteriores.

Uso de **GetBucketInventoryConfiguration** con un AWS SDK o la CLI

Los siguientes ejemplos de código muestran cómo utilizar `GetBucketInventoryConfiguration`.

CLI

AWS CLI

Recuperar la configuración de inventario de un bucket

En el siguiente ejemplo de `get-bucket-inventory-configuration`, se recupera la configuración de inventario del bucket especificado con el ID 1.

```
aws s3api get-bucket-inventory-configuration \
    --bucket my-bucket \
    --id 1
```

Salida:

```
{
    "InventoryConfiguration": {
        "IsEnabled": true,
        "Destination": {
            "S3BucketDestination": {
                "Format": "ORC",
                "Bucket": "arn:aws:s3:::my-bucket",
                "AccountId": "123456789012"
            }
        },
        "IncludedObjectVersions": "Current",
        "Id": "1",
        "Schedule": {
            "Frequency": "Weekly"
        }
    }
}
```

- Para obtener información sobre la API, consulte GetBucketInventoryConfiguration en la Referencia de comandos de la AWS CLI.

PowerShell

Herramientas para PowerShell

Ejemplo 1: este comando devuelve los detalles del inventario con el nombre “testinventory” del bucket de S3 indicado.

```
Get-S3BucketInventoryConfiguration -BucketName 's3testbucket' -InventoryId
 'testinventory'
```

- Para obtener información sobre la API, consulte GetBucketInventoryConfiguration en la Referencia de Cmdlet de AWS Tools for PowerShell.

Para obtener una lista completa de las guías para desarrolladores del AWS SDK y ejemplos de código, consulte Uso de este servicio con un SDK de AWS. En este tema también se incluye información sobre cómo comenzar a utilizar el SDK y detalles sobre sus versiones anteriores.

Uso de **GetBucketLifecycleConfiguration** con un AWS SDK o la CLI

Los siguientes ejemplos de código muestran cómo utilizar `GetBucketLifecycleConfiguration`.

.NET

AWS SDK for .NET

> **Note**
>
> Hay más en GitHub. Busque el ejemplo completo y aprenda a configurar y ejecutar en el Repositorio de ejemplos de código de 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;
        }
```

- Para obtener información sobre la API, consulte GetBucketLifecycleConfiguration en la Referencia de la API de AWS SDK for .NET.

CLI

AWS CLI

El siguiente comando recupera la configuración del ciclo de vida de un bucket denominado `my-bucket`:

```
aws s3api get-bucket-lifecycle-configuration --bucket my-bucket
```

Salida:

```
{
    "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"
        }
    ]
}
```

- Para obtener información sobre la API, consulte GetBucketLifecycleConfiguration en la Referencia de comandos de la AWS CLI.

Python

SDK para Python (Boto3)

> **Note**
>
> Hay más en GitHub. Busque el ejemplo completo y aprenda a configurar y ejecutar en el Repositorio de ejemplos de código de 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
```

- Para obtener información sobre la API, consulte GetBucketLifecycleConfiguration en la Referencia de la API de AWS SDK para Python (Boto3).

Para obtener una lista completa de las guías para desarrolladores del AWS SDK y ejemplos de código, consulte Uso de este servicio con un SDK de AWS. En este tema también se incluye información sobre cómo comenzar a utilizar el SDK y detalles sobre sus versiones anteriores.

Uso de **`GetBucketLocation`** con un AWS SDK o la CLI

Los siguientes ejemplos de código muestran cómo utilizar `GetBucketLocation`.

CLI

AWS CLI

El siguiente comando recupera la restricción de ubicación de un bucket denominado `my-bucket`, si existe una restricción:

```
aws s3api get-bucket-location --bucket my-bucket
```

Salida:

```
{
    "LocationConstraint": "us-west-2"
}
```

- Para obtener información sobre la API, consulte GetBucketLocation en la Referencia de comandos de la AWS CLI.

PowerShell

Herramientas para PowerShell

Ejemplo 1: este comando devuelve la restricción de ubicación del bucket “s3testbucket”, si existe una restricción.

```
Get-S3BucketLocation -BucketName 's3testbucket'
```

Salida:

```
Value
-----
ap-south-1
```

- Para obtener información sobre la API, consulte GetBucketLocation en la Referencia de Cmdlet de AWS Tools for PowerShell.

Rust

SDK para Rust

> **Note**
>
> Hay más información en GitHub. Busque el ejemplo completo y aprenda a configurar y ejecutar en el Repositorio de ejemplos de código de 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(())
}
```

- Para obtener información sobre la API, consulte GetBucketLocation en la Referencia de la API de AWS SDK para Rust.

Para obtener una lista completa de las guías para desarrolladores del AWS SDK y ejemplos de código, consulte Uso de este servicio con un SDK de AWS. En este tema también se incluye información sobre cómo comenzar a utilizar el SDK y detalles sobre sus versiones anteriores.

Uso de **`GetBucketLogging`** con un AWS SDK o la CLI

Los siguientes ejemplos de código muestran cómo utilizar `GetBucketLogging`.

CLI

AWS CLI

Recuperar el estado de registros de un bucket

En el siguiente ejemplo de `get-bucket-logging`, se recupera el estado de registros del bucket especificado.

```
aws s3api get-bucket-logging \
    --bucket my-bucket
```

Salida:

```
{
    "LoggingEnabled": {
        "TargetPrefix": "",
        "TargetBucket": "my-bucket-logs"
          }
}
```

- Para obtener información sobre la API, consulte GetBucketLogging en la Referencia de comandos de la AWS CLI.

PowerShell

Herramientas para PowerShell

Ejemplo 1: este comando devuelve el estado de registros del bucket especificado.

```
Get-S3BucketLogging -BucketName 's3testbucket'
```

Salida:

```
TargetBucketName   Grants TargetPrefix
----------------   ------ ------------
testbucket1        {}     testprefix
```

- Para obtener información sobre la API, consulte GetBucketLogging en la Referencia de Cmdlet de AWS Tools for PowerShell.

Para obtener una lista completa de las guías para desarrolladores del AWS SDK y ejemplos de código, consulte Uso de este servicio con un SDK de AWS. En este tema también se incluye información sobre cómo comenzar a utilizar el SDK y detalles sobre sus versiones anteriores.

Uso de **GetBucketMetricsConfiguration** con un AWS SDK o la CLI

Los siguientes ejemplos de código muestran cómo utilizar `GetBucketMetricsConfiguration`.

CLI

AWS CLI

Recuperar la configuración de métricas de un bucket con un ID específico

En el siguiente ejemplo de `get-bucket-metrics-configuration`, se muestra la configuración de métricas para el bucket e ID especificados.

```
aws s3api get-bucket-metrics-configuration \
    --bucket my-bucket \
    --id 123
```

Salida:

```
{
```

```
    "MetricsConfiguration": {
        "Filter": {
            "Prefix": "logs"
        },
        "Id": "123"
    }
}
```

- Para obtener información sobre la API, consulte GetBucketMetricsConfiguration en la Referencia de comandos de la AWS CLI.

PowerShell

Herramientas para PowerShell

Ejemplo 1: este comando devuelve los detalles del filro de métricas con el nombre “testfilter” del bucket de S3 indicado.

```
Get-S3BucketMetricsConfiguration -BucketName 's3testbucket' -MetricsId
 'testfilter'
```

- Para obtener información sobre la API, consulte GetBucketMetricsConfiguration en la Referencia de Cmdlet de AWS Tools for PowerShell.

Para obtener una lista completa de las guías para desarrolladores del AWS SDK y ejemplos de código, consulte Uso de este servicio con un SDK de AWS. En este tema también se incluye información sobre cómo comenzar a utilizar el SDK y detalles sobre sus versiones anteriores.

Uso de **`GetBucketNotification`** con un AWS SDK o la CLI

Los siguientes ejemplos de código muestran cómo utilizar `GetBucketNotification`.

CLI

AWS CLI

El siguiente comando recupera la configuración de notificaciones de un bucket denominado `my-bucket`:

```
aws s3api get-bucket-notification --bucket my-bucket
```

Salida:

```
{
    "TopicConfiguration": {
        "Topic": "arn:aws:sns:us-west-2:123456789012:my-notification-topic",
        "Id": "YmQzMmEwM2EjZWVlI0NGItNzVtZjI1MC00ZjgyLWZDBiZWNl",
        "Event": "s3:ObjectCreated:*",
        "Events": [
            "s3:ObjectCreated:*"
        ]
    }
}
```

- Para obtener información sobre la API, consulte GetBucketNotification en la Referencia de comandos de la AWS CLI.

PowerShell

Herramientas para PowerShell

Ejemplo 1: en este ejemplo se recupera la configuración de notificaciones del bucket en cuestión

```
Get-S3BucketNotification -BucketName kt-tools | select -ExpandProperty
 TopicConfigurations
```

Salida:

```
Id   Topic
--   -----
mimo arn:aws:sns:eu-west-1:123456789012:topic-1
```

- Para obtener información sobre la API, consulte GetBucketNotification en la Referencia de Cmdlet de AWS Tools for PowerShell.

Para obtener una lista completa de las guías para desarrolladores del AWS SDK y ejemplos de código, consulte Uso de este servicio con un SDK de AWS. En este tema también se incluye información sobre cómo comenzar a utilizar el SDK y detalles sobre sus versiones anteriores.

Uso de **`GetBucketPolicy`** con un AWS SDK o la CLI

Los siguientes ejemplos de código muestran cómo utilizar `GetBucketPolicy`.

C++

SDK para C++

> **Note**
>
> Hay más en GitHub. Busque el ejemplo completo y aprenda a configurar y ejecutar en el Repositorio de ejemplos de código de 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();
}
```

- Para obtener información sobre la API, consulte GetBucketPolicy en la Referencia de la API de AWS SDK for C++.

CLI

AWS CLI

El siguiente comando recupera la política de bucket de un bucket denominado `my-bucket`:

```
aws s3api get-bucket-policy --bucket my-bucket
```

Salida:

```
{
    "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/*
\"}]}"
}
```

Obtención y colocación de una política de bucket En el siguiente ejemplo se muestra cómo se puede descargar una política de bucket de Amazon S3, realizar modificaciones en el archivo y luego usar `put-bucket-policy` para aplicar la política de bucket modificada. Para descargar la política de bucket a un archivo, puede ejecutar:

```
aws s3api get-bucket-policy --bucket mybucket --query Policy --output text >
 policy.json
```

A continuación, puede modificar el archivo `policy.json` según sea necesario. Por último, puede volver a aplicar esta política modificada al bucket de S3 ejecutando:

archivo `policy.json` según sea necesario. Por último, puede volver a aplicar esta política modificada al bucket de S3 ejecutando:

archivo según sea necesario. Por último, puede volver a aplicar esta política modificada al bucket de S3 ejecutando:

```
aws s3api put-bucket-policy --bucket mybucket --policy file://policy.json
```

- Para obtener información sobre la API, consulte GetBucketPolicy en la Referencia de comandos de la AWS CLI.

Java

SDK para Java 2.x

> **Note**
>
> Hay más en GitHub. Busque el ejemplo completo y aprenda a configurar y ejecutar en el Repositorio de ejemplos de código de 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 "";
    }
}
```

- Para obtener información sobre la API, consulte GetBucketPolicy en la Referencia de la API de AWS SDK for Java 2.x.

JavaScript

SDK para JavaScript (v3)

> **Note**
>
> Hay más información en GitHub. Busque el ejemplo completo y aprenda a configurar y ejecutar en el Repositorio de ejemplos de código de AWS.

Obtenga la política del bucket.

```
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);
  }
};
```

- Para obtener información, consulte la Guía para desarrolladores de AWS SDK for JavaScript.
- Para obtener información sobre la API, consulte GetBucketPolicy en la Referencia de la API de AWS SDK for JavaScript.

Kotlin

SDK para Kotlin

> **Note**
>
> Hay más información en GitHub. Busque el ejemplo completo y aprenda a configurar y ejecutar en el Repositorio de ejemplos de código de 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
    }
}
```

- Para obtener información sobre la API, consulte GetBucketPolicy en la Referencia de la API de AWS SDK para Kotlin.

PowerShell

Herramientas para PowerShell

Ejemplo 1: este comando muestra la política de bucket asociada al bucket de S3 indicado.

```
Get-S3BucketPolicy -BucketName 's3testbucket'
```

- Para obtener información sobre la API, consulte GetBucketPolicy en la Referencia de Cmdlet de AWS Tools for PowerShell.

Python

SDK para Python (Boto3)

> **Note**
>
> Hay más en GitHub. Busque el ejemplo completo y aprenda a configurar y ejecutar en el Repositorio de ejemplos de código de 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)
```

- Para obtener información sobre la API, consulte GetBucketPolicy en la Referencia de la API de AWS SDK para Python (Boto3).

Ruby

SDK para Ruby

> **Note**
>
> Hay más en GitHub. Busque el ejemplo completo y aprenda a configurar y ejecutar en el Repositorio de ejemplos de código de 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
```

- Para obtener información sobre la API, consulte GetBucketPolicy en la Referencia de la API de AWS SDK for Ruby.

Para obtener una lista completa de las guías para desarrolladores del AWS SDK y ejemplos de código, consulte Uso de este servicio con un SDK de AWS. En este tema también se incluye información sobre cómo comenzar a utilizar el SDK y detalles sobre sus versiones anteriores.

Uso de **GetBucketPolicyStatus** con un AWS SDK o la CLI

Los siguientes ejemplos de código muestran cómo utilizar `GetBucketPolicyStatus`.

CLI

AWS CLI

Recuperar el estado de política de un bucket que indica si el bucket es público

En el siguiente ejemplo de `get-bucket-policy-status`, se recupera el estado de política del bucket `my-bucket`.

```
aws s3api get-bucket-policy-status \
    --bucket my-bucket
```

Salida:

```
{
    "PolicyStatus": {
        "IsPublic": false
    }
}
```

- Para obtener información sobre la API, consulte GetBucketPolicyStatus en la Referencia de comandos de la AWS CLI.

PowerShell

Herramientas para PowerShell

Ejemplo 1: este comando devuelve el estado de política del bucket específico de S3 e indica si el bucket es público.

```
Get-S3BucketPolicyStatus -BucketName 's3casetestbucket'
```

- Para obtener información sobre la API, consulte GetBucketPolicyStatus en la Referencia de Cmdlet de AWS Tools for PowerShell.

Para obtener una lista completa de las guías para desarrolladores del AWS SDK y ejemplos de código, consulte Uso de este servicio con un SDK de AWS. En este tema también se incluye información sobre cómo comenzar a utilizar el SDK y detalles sobre sus versiones anteriores.

Uso de **GetBucketReplication** con un AWS SDK o la CLI

Los siguientes ejemplos de código muestran cómo utilizar `GetBucketReplication`.

CLI

AWS CLI

El siguiente comando recupera la configuración de replicación de un bucket denominado `my-bucket`:

```
aws s3api get-bucket-replication --bucket my-bucket
```

Salida:

```
{
    "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"
    }
}
```

- Para obtener información sobre la API, consulte GetBucketReplication en la Referencia de comandos de la AWS CLI.

PowerShell

Herramientas para PowerShell

Ejemplo 1: devuelve la información de configuración de replicación establecida en el bucket denominado “mybucket”..

```
Get-S3BucketReplication -BucketName mybucket
```

- Para obtener información sobre la API, consulte GetBucketReplication en la Referencia de Cmdlet de AWS Tools for PowerShell.

Para obtener una lista completa de las guías para desarrolladores del AWS SDK y ejemplos de código, consulte Uso de este servicio con un SDK de AWS. En este tema también se incluye información sobre cómo comenzar a utilizar el SDK y detalles sobre sus versiones anteriores.

Uso de **`GetBucketRequestPayment`** con un AWS SDK o la CLI

Los siguientes ejemplos de código muestran cómo utilizar `GetBucketRequestPayment`.

CLI

AWS CLI

Recuperar la configuración de pagos de solicitudes de un bucket

En el siguiente ejemplo de `get-bucket-request-payment`, se recupera la configuración de pagos por el solicitante para el bucket especificado.

```
aws s3api get-bucket-request-payment \
    --bucket my-bucket
```

Salida:

```
{
    "Payer": "BucketOwner"
}
```

- Para obtener información sobre la API, consulte GetBucketRequestPayment en la Referencia de comandos de la AWS CLI.

PowerShell

Herramientas para PowerShell

Ejemplo 1: devuelve la configuración de pagos de solicitudes del bucket denominado “mybucket”.. De forma predeterminada, el propietario del bucket paga las descargas realizadas desde el bucket.

```
Get-S3BucketRequestPayment -BucketName mybucket
```

- Para obtener información sobre la API, consulte GetBucketRequestPayment en la Referencia de Cmdlet de AWS Tools for PowerShell.

Para obtener una lista completa de las guías para desarrolladores del AWS SDK y ejemplos de código, consulte Uso de este servicio con un SDK de AWS. En este tema también se incluye información sobre cómo comenzar a utilizar el SDK y detalles sobre sus versiones anteriores.

Uso de **`GetBucketTagging`** con un AWS SDK o la CLI

Los siguientes ejemplos de código muestran cómo utilizar `GetBucketTagging`.

CLI

AWS CLI

El siguiente comando recupera la configuración de etiquetado de un bucket denominado `my-bucket`:

```
aws s3api get-bucket-tagging --bucket my-bucket
```

Salida:

```
{
    "TagSet": [
        {
            "Value": "marketing",
            "Key": "organization"
        }
    ]
}
```

- Para obtener información sobre la API, consulte GetBucketTagging en la Referencia de comandos de la AWS CLI.

PowerShell

Herramientas para PowerShell

Ejemplo 1: este comando devuelve todas las etiquetas asociadas al bucket indicado.

```
Get-S3BucketTagging -BucketName 's3casetestbucket'
```

- Para obtener información sobre la API, consulte GetBucketTagging en la Referencia de Cmdlet de AWS Tools for PowerShell.

Para obtener una lista completa de las guías para desarrolladores del AWS SDK y ejemplos de código, consulte Uso de este servicio con un SDK de AWS. En este tema también se incluye información sobre cómo comenzar a utilizar el SDK y detalles sobre sus versiones anteriores.

Uso de **`GetBucketVersioning`** con un AWS SDK o la CLI

Los siguientes ejemplos de código muestran cómo utilizar `GetBucketVersioning`.

CLI

AWS CLI

El siguiente comando recupera la configuración del control de versiones de un bucket denominado `my-bucket`:

```
aws s3api get-bucket-versioning --bucket my-bucket
```

Salida:

```
{
    "Status": "Enabled"
}
```

- Para obtener información sobre la API, consulte GetBucketVersioning en la Referencia de comandos de la AWS CLI.

PowerShell

Herramientas para PowerShell

Ejemplo 1: este comando devuelve el estado del control de versiones con respecto al bucket indicado.

```
Get-S3BucketVersioning -BucketName 's3testbucket'
```

- Para obtener información sobre la API, consulte GetBucketVersioning en la Referencia de Cmdlet de AWS Tools for PowerShell.

Para obtener una lista completa de las guías para desarrolladores del AWS SDK y ejemplos de código, consulte Uso de este servicio con un SDK de AWS. En este tema también se incluye información sobre cómo comenzar a utilizar el SDK y detalles sobre sus versiones anteriores.

Uso de **`GetBucketWebsite`** con un AWS SDK o la CLI

Los siguientes ejemplos de código muestran cómo utilizar `GetBucketWebsite`.

.NET

AWS SDK for .NET

> **Note**
>
> Hay más en GitHub. Busque el ejemplo completo y aprenda a configurar y ejecutar en el Repositorio de ejemplos de código de 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}");
```

- Para obtener información sobre la API, consulte GetBucketWebsite en la Referencia de la API de AWS SDK for .NET.

C++

SDK para C++

> **Note**
>
> Hay más en GitHub. Busque el ejemplo completo y aprenda a configurar y ejecutar en el Repositorio de ejemplos de código de 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();
}
```

- Para obtener información sobre la API, consulte GetBucketWebsite en la Referencia de la API de AWS SDK for C++.

CLI

AWS CLI

El siguiente comando recupera la configuración de sitio web estática de un bucket denominado `my-bucket`:

```
aws s3api get-bucket-website --bucket my-bucket
```

Salida:

```
{
    "IndexDocument": {
        "Suffix": "index.html"
    },
    "ErrorDocument": {
        "Key": "error.html"
    }
}
```

- Para obtener información sobre la API, consulte GetBucketWebsite en la Referencia de comandos de la AWS CLI.

JavaScript

SDK para JavaScript (v3)

> **Note**
>
> Hay más información en GitHub. Busque el ejemplo completo y aprenda a configurar y ejecutar en el Repositorio de ejemplos de código de AWS.

Obtenga la configuración de sitio 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);
  }
};
```

- Para obtener información sobre la API, consulte GetBucketWebsite en la Referencia de la API de AWS SDK for JavaScript.

PowerShell

Herramientas para PowerShell

Ejemplo 1: este comando devuelve los detalles de las configuraciones de sitio web estáticas del bucket de S3 indicado.

```
Get-S3BucketWebsite -BucketName 's3testbucket'
```

- Para obtener información sobre la API, consulte GetBucketWebsite en la Referencia de Cmdlet de AWS Tools for PowerShell.

Para obtener una lista completa de las guías para desarrolladores del AWS SDK y ejemplos de código, consulte Uso de este servicio con un SDK de AWS. En este tema también se incluye información sobre cómo comenzar a utilizar el SDK y detalles sobre sus versiones anteriores.

Uso de **`GetObject`** con un AWS SDK o la CLI

Los siguientes ejemplos de código muestran cómo utilizar `GetObject`.

Los ejemplos de acciones son extractos de código de programas más grandes y deben ejecutarse en contexto. Puede ver esta acción en su contexto en los siguientes ejemplos de código:

- Obtenga un objeto de un bucket si se ha modificado
- Obtención de un objeto desde un punto de acceso de varias regiones
- Introducción a los buckets y objetos
- Introducción al cifrado
- Realización de un seguimiento de cargas y descargas

.NET

AWS SDK for .NET

> **Note**
>
> Hay más información en GitHub. Busque el ejemplo completo y aprenda a configurar y ejecutar en el Repositorio de ejemplos de código de 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;
            }
        }
```

- Para obtener información sobre la API, consulte GetObject en la Referencia de la API de AWS SDK for .NET.

Bash

AWS CLI con script Bash

> **Note**
>
> Hay más información en GitHub. Busque el ejemplo completo y aprenda a configurar y ejecutar en el Repositorio de ejemplos de código de 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
}
```

- Para obtener información sobre la API, consulte GetObject en la Referencia de comandos de AWS CLI.

C++

SDK para C++

> **Note**
>
> Hay más en GitHub. Busque el ejemplo completo y aprenda a configurar y ejecutar en el Repositorio de ejemplos de código de 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();
}
```

- Para obtener información sobre la API, consulte GetObject en la Referencia de la API de AWS SDK for C++.

CLI

AWS CLI

En el siguiente ejemplo se utiliza el comando `get-object` para descargar un objeto de Amazon S3.

```
aws s3api get-object --bucket text-content --key dir/
my_images.tar.bz2 my_images.tar.bz2
```

Tenga en cuenta que el parámetro outfile se especifica sin un nombre de opción, como "--outfile". El nombre del archivo de salida debe ser el último parámetro del comando.

El siguiente ejemplo muestra el uso de `--range` para descargar un intervalo de bytes específico de un objeto. Tenga en cuenta que los intervalos de bytes deben tener el prefijo "bytes=":

```
aws s3api get-object --bucket text-content --key dir/my_data --
range bytes=8888-9999 my_data_range
```

Para obtener más información acerca de la recuperación de objetos, consulte Obtención de objetos en la Guía para desarrolladores de Amazon S3.

- Para obtener detalles de la API, consulte GetObject en la Referencia de comandos de AWS CLI.

Go

SDK para Go V2

> **Note**
>
> Hay más en GitHub. Busque el ejemplo completo y aprenda a configurar y ejecutar en el Repositorio de ejemplos de código de 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
}
```

- Para obtener información sobre la API, consulte GetObject en la Referencia de la API de AWS SDK for Go.

Java

SDK para Java 2.x

> **Note**
>
> Hay más en GitHub. Busque el ejemplo completo y aprenda a configurar y ejecutar en el Repositorio de ejemplos de código de AWS.

Lea datos como una matriz de bytes con un S3 Client.

```
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);
        }
    }
}
```

Utilice un S3TransferManager para descargar un objeto de un bucket de S3 en un archivo local. Vea el archivo completo y pruébelo.

```
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();
    }
```

Lea las etiquetas que pertenecen a un objeto con 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);
        }
    }
}
```

Obtenga una URL para un objeto con 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);
        }
    }
}
```

Obtenga un objeto mediante el objeto de cliente S3Presigner con 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();
        }
    }
```

```
}
```

Obtenga un objeto mediante un objeto ResponseTransformer y 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);
        }
    }
}
```

- Para obtener información sobre la API, consulte GetObject en la Referencia de la API de AWS SDK for Java 2.x.

JavaScript

SDK para JavaScript (v3)

> **Note**
>
> Hay más información en GitHub. Busque el ejemplo completo y aprenda a configurar y ejecutar en el Repositorio de ejemplos de código de AWS.

Descargue el objeto.

```
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);
  }
};
```

- Para obtener información, consulte la Guía para desarrolladores de AWS SDK for JavaScript.
- Para obtener información sobre la API, consulte GetObject en la Referencia de la API de AWS SDK for JavaScript.

Kotlin

SDK para Kotlin

> **Note**
>
> Hay más información en GitHub. Busque el ejemplo completo y aprenda a configurar y ejecutar en el Repositorio de ejemplos de código de 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")
        }
    }
}
```

- Para obtener información sobre la API, consulte GetObject en la Referencia de la API de SDK de AWS para Kotlin.

PHP

SDK para PHP

> **Note**
>
> Hay más en GitHub. Busque el ejemplo completo y aprenda a configurar y ejecutar en el Repositorio de ejemplos de código de AWS.

Obtenga un objeto.

```
        $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.");
        }
```

- Para obtener información sobre la API, consulte GetObject en la Referencia de la API de AWS SDK for PHP.

PowerShell

Herramientas para PowerShell

Ejemplo 1: este comando recupera el elemento “sample.txt” del bucket “test-files” y lo guarda en un archivo denominado “local-sample.txt” en la ubicación actual. No es necesario que el archivo “local-sample.txt” exista para poder llamar a este comando.

```
Read-S3Object -BucketName test-files -Key sample.txt -File local-sample.txt
```

Ejemplo 2: este comando recupera el directorio virtual “DIR” del bucket “test-files” y lo guarda en una carpeta denominada “Local-DIR” en la ubicación actual. No es necesario que la carpeta “Local-DIR” exista para poder llamar a este comando.

```
Read-S3Object -BucketName test-files -KeyPrefix DIR -Folder Local-DIR
```

Ejemplo 3: descarga todos los objetos cuyas claves terminan en “.json” de los buckets con “config” en el nombre del bucket a los archivos de la carpeta especificada. Las claves de objeto se utilizan para establecer los nombres de los archivos.

```
Get-S3Bucket | ? { $_.BucketName -like '*config*' } | Get-S3Object | ? { $_.Key -
like '*.json' } | Read-S3Object -Folder C:\ConfigObjects
```

- Para obtener información sobre la API, consulte GetObject en la Referencia de Cmdlet de AWS Tools for PowerShell.

Python

SDK para Python (Boto3)

> **Note**
>
> Hay más en GitHub. Busque el ejemplo completo y aprenda a configurar y ejecutar en el Repositorio de ejemplos de código de 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
```

- Para obtener información sobre la API, consulte GetObject en la Referencia de la API de AWS SDK para Python (Boto3).

Ruby

SDK para Ruby

> **Note**
>
> Hay más en GitHub. Busque el ejemplo completo y aprenda a configurar y ejecutar en el Repositorio de ejemplos de código de AWS.

Obtenga un objeto.

```
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__
```

Obtenga un objeto e informe de su estado de cifrado del lado del servidor.

```
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__
```

- Para obtener información sobre la API, consulte GetObject en la Referencia de la API de AWS SDK for Ruby.

Rust

SDK para Rust

> **Note**
>
> Hay más información en GitHub. Busque el ejemplo completo y aprenda a configurar y ejecutar en el Repositorio de ejemplos de código de 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)
}
```

- Para obtener información sobre la API, consulte GetObject en la Referencia de la API de AWS SDK para Rust.

SAP ABAP

SDK de SAP ABAP

> **Note**
>
> Hay más en GitHub. Busque el ejemplo completo y aprenda a configurar y ejecutar en el Repositorio de ejemplos de código de 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.
```

- Para obtener información sobre la API, consulte GetObject en la Referencia de la API de AWS SDK para SAP ABAP.

Swift

SDK para Swift

> **Note**
>
> Esto es documentación preliminar para un SDK en versión preliminar. Está sujeta a cambios.

> **Note**
>
> Hay más en GitHub. Busque el ejemplo completo y aprenda a configurar y ejecutar en el Repositorio de ejemplos de código de AWS.

Descargar un objeto desde un bucket a un archivo 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)
    }
```

Lea un objeto en un objeto de datos de Swift.

```
    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
    }
```

- Para obtener información acerca de la API, consulte GetObject en la Referencia de la API del SDK de AWS para Swift.

Para obtener una lista completa de las guías para desarrolladores del AWS SDK y ejemplos de código, consulte Uso de este servicio con un SDK de AWS. En este tema también se incluye información sobre cómo comenzar a utilizar el SDK y detalles sobre sus versiones anteriores.

Uso de **`GetObjectAcl`** con un AWS SDK o la CLI

Los siguientes ejemplos de código muestran cómo utilizar `GetObjectAcl`.

Los ejemplos de acciones son extractos de código de programas más grandes y deben ejecutarse en contexto. Puede ver esta acción en su contexto en el siguiente ejemplo de código:

- Administrar listas de control de acceso (ACL)

C++

SDK para C++

> **Note**
>
> Hay más en GitHub. Busque el ejemplo completo y aprenda a configurar y ejecutar en el Repositorio de ejemplos de código de 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";
    }
}
```

- Para obtener información sobre la API, consulte GetObjectAcl en la Referencia de la API de AWS SDK for C++.

CLI

AWS CLI

El siguiente comando recupera la lista de control de acceso de un objeto en un bucket denominado `my-bucket`:

```
aws s3api get-object-acl --bucket my-bucket --key index.html
```

Salida:

```
{
    "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"
        }
    ]
}
```

- Para obtener información sobre la API, consulte GetObjectAcl en la Referencia de comandos de la AWS CLI.

Kotlin

SDK para Kotlin

> **Note**
>
> Hay más información en GitHub. Busque el ejemplo completo y aprenda a configurar y ejecutar en el Repositorio de ejemplos de código de 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}")
        }
    }
}
```

- Para obtener información sobre la API, consulte GetObjectAcl en la Referencia de la API de AWS SDK para Kotlin.

Python

SDK para Python (Boto3)

> **Note**
>
> Hay más en GitHub. Busque el ejemplo completo y aprenda a configurar y ejecutar en el Repositorio de ejemplos de código de 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
```

- Para obtener información sobre la API, consulte GetObjectAcl en la Referencia de la API de AWS SDK para Python (Boto3).

Para obtener una lista completa de las guías para desarrolladores del AWS SDK y ejemplos de código, consulte Uso de este servicio con un SDK de AWS. En este tema también se incluye información sobre cómo comenzar a utilizar el SDK y detalles sobre sus versiones anteriores.

Uso de **`GetObjectAttributes`** con un AWS SDK o la CLI

Los siguientes ejemplos de código muestran cómo utilizar `GetObjectAttributes`.

Los ejemplos de acciones son extractos de código de programas más grandes y deben ejecutarse en contexto. Puede ver esta acción en su contexto en el siguiente ejemplo de código:

- Trabajo con la integridad de los objetos de Amazon S3

C++

SDK para C++

> **Note**
>
> Hay más en GitHub. Busque el ejemplo completo y aprenda a configurar y ejecutar en el Repositorio de ejemplos de código de AWS.

```
// ! Routine which retrieves the hash value of an object stored in an S3 bucket.
/*!
   \param bucket: The name of the S3 bucket where the object is stored.
   \param key: The unique identifier (key) of the object within the S3 bucket.
   \param hashMethod: The hashing algorithm used to calculate the hash value of
 the object.
   \param[out] hashData: The retrieved hash.
```

```
   \param[out] partHashes: The part hashes if available.
   \param client: The S3 client instance used to retrieve the object.
   \return bool: Function succeeded.
*/
bool AwsDoc::S3::retrieveObjectHash(const Aws::String &bucket, const Aws::String
 &key,
                                    AwsDoc::S3::HASH_METHOD hashMethod,
                                    Aws::String &hashData,
                                    std::vector<Aws::String> *partHashes,
                                    const Aws::S3::S3Client &client) {
   Aws::S3::Model::GetObjectAttributesRequest request;
   request.SetBucket(bucket);
   request.SetKey(key);

   if (hashMethod == MD5) {
       Aws::Vector<Aws::S3::Model::ObjectAttributes> attributes;
       attributes.push_back(Aws::S3::Model::ObjectAttributes::ETag);
       request.SetObjectAttributes(attributes);

       Aws::S3::Model::GetObjectAttributesOutcome outcome =
 client.GetObjectAttributes(
               request);
       if (outcome.IsSuccess()) {
           const Aws::S3::Model::GetObjectAttributesResult &result =
 outcome.GetResult();
           hashData = result.GetETag();
       } else {
           std::cerr << "Error retrieving object etag attributes." <<
                     outcome.GetError().GetMessage() << std::endl;
           return false;
       }
   } else { // hashMethod != MD5
       Aws::Vector<Aws::S3::Model::ObjectAttributes> attributes;
       attributes.push_back(Aws::S3::Model::ObjectAttributes::Checksum);
       request.SetObjectAttributes(attributes);

       Aws::S3::Model::GetObjectAttributesOutcome outcome =
 client.GetObjectAttributes(
               request);
       if (outcome.IsSuccess()) {
           const Aws::S3::Model::GetObjectAttributesResult &result =
 outcome.GetResult();
           switch (hashMethod) {
               case AwsDoc::S3::DEFAULT: // NOLINT(*-branch-clone)
```

```
                    break;  // Default is not supported.
#pragma clang diagnostic push
#pragma ide diagnostic ignored "UnreachableCode"
                case AwsDoc::S3::MD5:
                    break;  // MD5 is not supported.
#pragma clang diagnostic pop
                case AwsDoc::S3::SHA1:
                    hashData = result.GetChecksum().GetChecksumSHA1();
                    break;
                case AwsDoc::S3::SHA256:
                    hashData = result.GetChecksum().GetChecksumSHA256();
                    break;
                case AwsDoc::S3::CRC32:
                    hashData = result.GetChecksum().GetChecksumCRC32();
                    break;
                case AwsDoc::S3::CRC32C:
                    hashData = result.GetChecksum().GetChecksumCRC32C();
                    break;
                default:
                    std::cerr << "Unknown hash method." << std::endl;
                    return false;
            }
        } else {
            std::cerr << "Error retrieving object checksum attributes." <<
                      outcome.GetError().GetMessage() << std::endl;
            return false;
        }

        if (nullptr != partHashes) {
            attributes.clear();
            attributes.push_back(Aws::S3::Model::ObjectAttributes::ObjectParts);
            request.SetObjectAttributes(attributes);
            outcome = client.GetObjectAttributes(request);
            if (outcome.IsSuccess()) {
                const Aws::S3::Model::GetObjectAttributesResult &result =
 outcome.GetResult();
                const Aws::Vector<Aws::S3::Model::ObjectPart> parts =
 result.GetObjectParts().GetParts();
                for (const Aws::S3::Model::ObjectPart &part: parts) {
                    switch (hashMethod) {
                        case AwsDoc::S3::DEFAULT: // Default is not supported.
 NOLINT(*-branch-clone)
                            break;
                        case AwsDoc::S3::MD5: // MD5 is not supported.
```

```
                            break;
                        case AwsDoc::S3::SHA1:
                            partHashes->push_back(part.GetChecksumSHA1());
                            break;
                        case AwsDoc::S3::SHA256:
                            partHashes->push_back(part.GetChecksumSHA256());
                            break;
                        case AwsDoc::S3::CRC32:
                            partHashes->push_back(part.GetChecksumCRC32());
                            break;
                        case AwsDoc::S3::CRC32C:
                            partHashes->push_back(part.GetChecksumCRC32C());
                            break;
                        default:
                            std::cerr << "Unknown hash method." << std::endl;
                            return false;
                    }
                }
            } else {
                std::cerr << "Error retrieving object attributes for object
 parts." <<
                          outcome.GetError().GetMessage() << std::endl;
                return false;
            }
        }
    }

    return true;
}
```

- Para obtener información sobre la API, consulte GetObjectAttributes en la Referencia de la API de AWS SDK for C++.

CLI

AWS CLI

Para recuperar metadatos de un objeto sin devolver el objeto en sí

En el siguiente ejemplo de `get-object-attributes`, se recuperan los metadatos del objeto `doc1.rtf`.

```
aws s3api get-object-attributes \
    --bucket my-bucket \
    --key doc1.rtf \
    --object-attributes "StorageClass" "ETag" "ObjectSize"
```

Salida:

```
{
    "LastModified": "2022-03-15T19:37:31+00:00",
    "VersionId": "IuCPjXTDzHNfldAuitVBIKJpF2p1fg4P",
    "ETag": "b662d79adeb7c8d787ea7eafb9ef6207",
    "StorageClass": "STANDARD",
    "ObjectSize": 405
}
```

Para obtener más información, consulte GetObjectAttributes en la Referencia de la API de Amazon S3.

- Para obtener información sobre la API, consulte GetObjectAttributes en la Referencia de comandos de la AWS CLI.

Para obtener una lista completa de las guías para desarrolladores del AWS SDK y ejemplos de código, consulte Uso de este servicio con un SDK de AWS. En este tema también se incluye información sobre cómo comenzar a utilizar el SDK y detalles sobre sus versiones anteriores.

Uso de **GetObjectLegalHold** con un AWS SDK o la CLI

Los siguientes ejemplos de código muestran cómo utilizar `GetObjectLegalHold`.

Los ejemplos de acciones son extractos de código de programas más grandes y deben ejecutarse en contexto. Puede ver esta acción en su contexto en los siguientes ejemplos de código:

- Obtención de la configuración de retención legal de un objeto
- Bloqueo de objetos de Amazon S3

.NET

AWS SDK for .NET

> **Note**
>
> Hay más información en GitHub. Busque el ejemplo completo y aprenda a configurar y ejecutar en el Repositorio de ejemplos de código de 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();
        }
    }
```

- Para obtener información sobre la API, consulte GetObjectLegalHold en la Referencia de la API de AWS SDK for .NET.

CLI

AWS CLI

Recupera el estado de retención legal de un objeto

En el siguiente ejemplo de `get-object-legal-hold`, se recupera el estado de retención legal del objeto especificado.

```
aws s3api get-object-legal-hold \
    --bucket my-bucket-with-object-lock \
    --key doc1.rtf
```

Salida:

```
{
    "LegalHold": {
        "Status": "ON"
    }
}
```

- Para obtener información sobre la API, consulte GetObjectLegalHold en la Referencia de comandos de la AWS CLI.

Go

SDK para Go V2

> **Note**
>
> Hay más en GitHub. Busque el ejemplo completo y aprenda a configurar y ejecutar en el Repositorio de ejemplos de código de 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
}
```

- Para obtener información sobre la API, consulte GetObjectLegalHold en la Referencia de la API de AWS SDK for Go.

Java

SDK para Java 2.x

> **Note**
>
> Hay más en GitHub. Busque el ejemplo completo y aprenda a configurar y ejecutar en el Repositorio de ejemplos de código de 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;
   }
```

- Para obtener información sobre la API, consulte GetObjectLegalHold en la Referencia de la API de AWS SDK for Java 2.x.

Python

SDK para Python (Boto3)

> **Note**
>
> Hay más en GitHub. Busque el ejemplo completo y aprenda a configurar y ejecutar en el Repositorio de ejemplos de código de AWS.

Ponga una retención legal en un objeto.

```
def get_legal_hold(s3_client, bucket: str, key: str) -> None:
    """
    Get the legal hold status of a specific file in a bucket.

    Args:
        s3_client: Boto3 S3 client.
        bucket: The name of the bucket containing the file.
        key: The key of the file to get the legal hold status of.
    """
    print()
    logger.info("Getting legal hold status of file [%s] in bucket [%s]", key,
 bucket)
    try:
        response = s3_client.get_object_legal_hold(Bucket=bucket, Key=key)
        legal_hold_status = response["LegalHold"]["Status"]
        logger.debug(
            "Legal hold status of file [%s] in bucket [%s] is [%s]",
            key,
            bucket,
            legal_hold_status,
        )
    except Exception as e:
        logger.error(
            "Failed to get legal hold status of file [%s] in bucket [%s]: %s",
            key,
            bucket,
            e,
        )
```

- Para obtener información sobre la API, consulte GetObjectLegalHold en la Referencia de la API de AWS SDK para Python (Boto3).

Para obtener una lista completa de las guías para desarrolladores del AWS SDK y ejemplos de código, consulte Uso de este servicio con un SDK de AWS. En este tema también se incluye información sobre cómo comenzar a utilizar el SDK y detalles sobre sus versiones anteriores.

Uso de **`GetObjectLockConfiguration`** con un AWS SDK o la CLI

Los siguientes ejemplos de código muestran cómo utilizar `GetObjectLockConfiguration`.

Los ejemplos de acciones son extractos de código de programas más grandes y deben ejecutarse en contexto. Puede ver esta acción en contexto en el siguiente ejemplo de código:

- Bloqueo de objetos de Amazon S3

.NET

AWS SDK for .NET

> **Note**
>
> Hay más información en GitHub. Busque el ejemplo completo y aprenda a configurar y ejecutar en el Repositorio de ejemplos de código de 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();
        }
    }
```

- Para obtener información sobre la API, consulte GetObjectLockConfiguration en la Referencia de la API de AWS SDK for .NET..

CLI

AWS CLI

Para recuperar una configuración de bloqueo de objetos para un bucket

En el siguiente ejemplo de `get-object-lock-configuration`, se recupera la configuración de bloqueo de objetos para el bucket especificado.

```
aws s3api get-object-lock-configuration \
    --bucket my-bucket-with-object-lock
```

Salida:

```
{
    "ObjectLockConfiguration": {
```

```
        "ObjectLockEnabled": "Enabled",
        "Rule": {
            "DefaultRetention": {
                "Mode": "COMPLIANCE",
                "Days": 50
            }
        }
    }
}
```

- Para obtener información sobre la API, consulte GetObjectLockConfiguration en la Referencia de comandos de la AWS CLI.

Go

SDK para Go V2

> **Note**
>
> Hay más en GitHub. Busque el ejemplo completo y aprenda a configurar y ejecutar en el Repositorio de ejemplos de código de 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
}
```

- Para obtener información sobre la API, consulte GetObjectLockConfiguration en la Referencia de la API de AWS SDK for Go..

Java

SDK para Java 2.x

> **Note**
>
> Hay más en GitHub. Busque el ejemplo completo y aprenda a configurar y ejecutar en el Repositorio de ejemplos de código de 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());
    }
```

- Para obtener información sobre la API, consulte GetObjectLockConfiguration en la Referencia de la API de AWS SDK for Java 2.x..

JavaScript

SDK para JavaScript (v3)

> **Note**
>
> Hay más información en GitHub. Busque el ejemplo completo y aprenda a configurar y ejecutar en el Repositorio de ejemplos de código de 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");
}
```

- Para obtener información sobre la API, consulte GetObjectLockConfiguration en la Referencia de la API de AWS SDK for JavaScript..

PowerShell

Herramientas para PowerShell

Ejemplo 1: este comando devuelve el valor “Enabled” si la configuración de bloqueo de objetos está habilitada para el bucket de S3 indicado.

```
Get-S3ObjectLockConfiguration -BucketName 's3buckettesting' -Select
 ObjectLockConfiguration.ObjectLockEnabled
```

Salida:

```
Value
-----
Enabled
```

- Para obtener información sobre la API, consulte GetObjectLockConfiguration en la Referencia de Cmdlet de AWS Tools for PowerShell.

Python

SDK para Python (Boto3)

> **Note**
>
> Hay más en GitHub. Busque el ejemplo completo y aprenda a configurar y ejecutar en el Repositorio de ejemplos de código de AWS.

Obtenga la configuración de bloqueo de objetos.

```
def is_object_lock_enabled(s3_client, bucket: str) -> bool:
    """
    Check if object lock is enabled for a bucket.

    Args:
        s3_client: Boto3 S3 client.
        bucket: The name of the bucket to check.

    Returns:
        True if object lock is enabled, False otherwise.
    """
    try:
        response = s3_client.get_object_lock_configuration(Bucket=bucket)
        return (
            "ObjectLockConfiguration" in response
            and response["ObjectLockConfiguration"]["ObjectLockEnabled"] ==
 "Enabled"
        )
    except s3_client.exceptions.ClientError as e:
        if e.response["Error"]["Code"] == "ObjectLockConfigurationNotFoundError":
            return False
        else:
            raise
```

- Para obtener información sobre la API, consulte GetObjectLockConfiguration en la Referencia de la del SDK de AWS SDK para Python (Boto3).

Para obtener una lista completa de las guías para desarrolladores del AWS SDK y ejemplos de código, consulte Uso de este servicio con un SDK de AWS. En este tema también se incluye información sobre cómo comenzar a utilizar el SDK y detalles sobre sus versiones anteriores.

Uso de **`GetObjectRetention`** con un AWS SDK o la CLI

Los siguientes ejemplos de código muestran cómo utilizar `GetObjectRetention`.

Los ejemplos de acciones son extractos de código de programas más grandes y deben ejecutarse en contexto. Puede ver esta acción en contexto en el siguiente ejemplo de código:

- Bloqueo de objetos de Amazon S3

.NET

AWS SDK for .NET

> **Note**
>
> Hay más información en GitHub. Busque el ejemplo completo y aprenda a configurar y ejecutar en el Repositorio de ejemplos de código de 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();
        }
    }
```

- Para ver la información de la API, consulte GetObjectRetention en la Referencia de la API de AWS SDK for .NET.

CLI

AWS CLI

Para recuperar la configuración de retención de un objeto

En el siguiente ejemplo de `get-object-retention`, se recupera la configuración de retención del objeto especificado.

```
aws s3api get-object-retention \
    --bucket my-bucket-with-object-lock \
    --key doc1.rtf
```

Salida:

```
{
    "Retention": {
        "Mode": "GOVERNANCE",
        "RetainUntilDate": "2025-01-01T00:00:00.000Z"
    }
}
```

- Para obtener información sobre la API, consulte GetObjectRetention en la Referencia de comandos de la AWS CLI.

Go

SDK para Go V2

> **Note**
>
> Hay más en GitHub. Busque el ejemplo completo y aprenda a configurar y ejecutar en el Repositorio de ejemplos de código de 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
}
```

- Para ver la información de la API, consulte GetObjectRetention en la Referencia de la API de AWS SDK for Go.

Java

SDK para Java 2.x

> **Note**
>
> Hay más en GitHub. Busque el ejemplo completo y aprenda a configurar y ejecutar en el Repositorio de ejemplos de código de 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;
        }
    }
```

- Para ver la información de la API, consulte GetObjectRetention en la Referencia de la API de AWS SDK for Java 2.x.

JavaScript

SDK para JavaScript (v3)

> **Note**
>
> Hay más información en GitHub. Busque el ejemplo completo y aprenda a configurar y ejecutar en el Repositorio de ejemplos de código de 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");
}
```

- Para ver la información de la API, consulte GetObjectRetention en la Referencia de la API de AWS SDK for JavaScript.

PowerShell

Herramientas para PowerShell

Ejemplo 1: el comando devuelve el modo y la fecha hasta que se retenga el objeto.

```
Get-S3ObjectRetention -BucketName 's3buckettesting' -Key 'testfile.txt'
```

- Para obtener información sobre la API, consulte GetObjectRetention en la Referencia de Cmdlet de AWS Tools for PowerShell.

Para obtener una lista completa de las guías para desarrolladores del AWS SDK y ejemplos de código, consulte Uso de este servicio con un SDK de AWS. En este tema también se incluye información sobre cómo comenzar a utilizar el SDK y detalles sobre sus versiones anteriores.

Uso de **`GetObjectTagging`** con un AWS SDK o la CLI

Los siguientes ejemplos de código muestran cómo utilizar `GetObjectTagging`.

Los ejemplos de acciones son extractos de código de programas más grandes y deben ejecutarse en contexto. Puede ver esta acción en contexto en el siguiente ejemplo de código:

- Comenzar a utilizar etiquetas

CLI

AWS CLI

Recuperar las etiquetas asociadas a un objeto

El siguiente ejemplo de `get-object-tagging` recupera los valores de la clave especificada del objeto especificado.

```
aws s3api get-object-tagging \
    --bucket my-bucket \
    --key doc1.rtf
```

Salida:

```
{
    "TagSet": [
        {
            "Value": "confidential",
            "Key": "designation"
        }
    ]
}
```

El siguiente ejemplo de `get-object-tagging` intenta recuperar los conjuntos de etiquetas del objeto `doc2.rtf`, que no tiene etiquetas.

```
aws s3api get-object-tagging \
    --bucket my-bucket \
    --key doc2.rtf
```

Salida:

```
{
    "TagSet": []
}
```

El siguiente ejemplo de `get-object-tagging` recupera los conjuntos de etiquetas del objeto `doc3.rtf`, que tiene varias etiquetas.

```
aws s3api get-object-tagging \
```

```
    --bucket my-bucket \
    --key doc3.rtf
```

Salida:

```
{
    "TagSet": [
        {
            "Value": "confidential",
            "Key": "designation"
        },
        {
            "Value": "finance",
            "Key": "department"
        },
        {
            "Value": "payroll",
            "Key": "team"
        }
    ]
}
```

- Para obtener información sobre la API, consulte GetObjectTagging en la Referencia de comandos de la AWS CLI.

PowerShell

Herramientas para PowerShell

Ejemplo 1: el ejemplo devuelve las etiquetas asociadas al objeto presentes en el bucket de S3 indicado.

```
Get-S3ObjectTagSet -Key 'testfile.txt' -BucketName 'testbucket123'
```

Salida:

```
Key  Value
---  -----
test value
```

- Para obtener información sobre la API, consulte GetObjectTagging en la Referencia de Cmdlet de AWS Tools for PowerShell.

Para obtener una lista completa de las guías para desarrolladores del AWS SDK y ejemplos de código, consulte Uso de este servicio con un SDK de AWS. En este tema también se incluye información sobre cómo comenzar a utilizar el SDK y detalles sobre sus versiones anteriores.

Uso de **GetPublicAccessBlock** con un AWS SDK o la CLI

Los siguientes ejemplos de código muestran cómo utilizar `GetPublicAccessBlock`.

CLI

AWS CLI

Establecer o modificar la configuración de bloqueo de acceso público de un bucket

En el siguiente ejemplo de `get-public-access-block`, se elimina la configuración de bloqueo de acceso público del bucket especificado.

```
aws s3api get-public-access-block \
    --bucket my-bucket
```

Salida:

```
{
    "PublicAccessBlockConfiguration": {
        "IgnorePublicAcls": true,
        "BlockPublicPolicy": true,
        "BlockPublicAcls": true,
        "RestrictPublicBuckets": true
    }
}
```

- Para obtener información sobre la API, consulte GetPublicAccessBlock en la Referencia de comandos de la AWS CLI.

PowerShell

Herramientas para PowerShell

Ejemplo 1: el comando devuelve la configuración de bloqueo de acceso público del bucket de S3 indicado.

```
Get-S3PublicAccessBlock -BucketName 's3testbucket'
```

- Para obtener información sobre la API, consulte GetPublicAccessBlock en la Referencia de Cmdlet de AWS Tools for PowerShell.

Para obtener una lista completa de las guías para desarrolladores del AWS SDK y ejemplos de código, consulte Uso de este servicio con un SDK de AWS. En este tema también se incluye información sobre cómo comenzar a utilizar el SDK y detalles sobre sus versiones anteriores.

Uso de **HeadBucket** con un AWS SDK o la CLI

Los siguientes ejemplos de código muestran cómo utilizar `HeadBucket`.

Bash

AWS CLI con script Bash

> **Note**
>
> Hay más información en GitHub. Busque el ejemplo completo y aprenda a configurar y ejecutar en el Repositorio de ejemplos de código de 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
}
```

- Para obtener información sobre la API, consulte HeadBucket en la Referencia de comandos de AWS CLI.

CLI

AWS CLI

El siguiente comando verifica el acceso a un bucket denominado `my-bucket`:

```
aws s3api head-bucket --bucket my-bucket
```

Si el bucket existe y tiene acceso a él, no se muestra ningún resultado. De lo contrario, se mostrará un mensaje de error. Por ejemplo:

```
A client error (404) occurred when calling the HeadBucket operation: Not Found
```

- Para obtener información sobre la API, consulte HeadBucket en la Referencia de comandos de AWS CLI.

Go

SDK para Go V2

> **Note**
>
> Hay más en GitHub. Busque el ejemplo completo y aprenda a configurar y ejecutar en el Repositorio de ejemplos de código de 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
}
```

- Para obtener información sobre la API, consulte HeadBucket en la Referencia de la API de AWS SDK for Go.

Python

SDK para Python (Boto3)

> **Note**
>
> Hay más en GitHub. Busque el ejemplo completo y aprenda a configurar y ejecutar en el Repositorio de ejemplos de código de 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
```

- Para obtener información sobre la API, consulte HeadBucket en la Referencia de la API de AWS SDK para Python (Boto3).

Para obtener una lista completa de las guías para desarrolladores del AWS SDK y ejemplos de código, consulte Uso de este servicio con un SDK de AWS. En este tema también se incluye información sobre cómo comenzar a utilizar el SDK y detalles sobre sus versiones anteriores.

Uso de **`HeadObject`** con un AWS SDK o la CLI

Los siguientes ejemplos de código muestran cómo utilizar `HeadObject`.

CLI

AWS CLI

El siguiente comando recupera los metadatos de un objeto de un bucket denominado `my-bucket`.

```
aws s3api head-object --bucket my-bucket --key index.html
```

Salida:

```
{
    "AcceptRanges": "bytes",
    "ContentType": "text/html",
    "LastModified": "Thu, 16 Apr 2015 18:19:14 GMT",
```

```
    "ContentLength": 77,
    "VersionId": "null",
    "ETag": "\"30a6ec7e1a9ad79c203d05a589c8b400\"",
    "Metadata": {}
}
```

- Para obtener información de la API, consulte HeadObject en la Referencia de comandos de AWS CLI.

Java

SDK para Java 2.x

> **Note**
>
> Hay más en GitHub. Busque el ejemplo completo y aprenda a configurar y ejecutar en el Repositorio de ejemplos de código de AWS.

Determinar el tipo de contenido de un objeto.

```
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);
        }
    }
}
```

Obtener el estado de restauración de un objeto.

```
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);
        }
    }
}
```

- Para obtener información sobre la API, consulte HeadObject en la Referencia de la API de AWS SDK for Java 2.x.

Ruby

SDK para Ruby

> **Note**
>
> Hay más en GitHub. Busque el ejemplo completo y aprenda a configurar y ejecutar en el Repositorio de ejemplos de código de 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__
```

- Para obtener información sobre la API, consulte HeadObject en la Referencia de la API de AWS SDK for Ruby.

Para obtener una lista completa de las guías para desarrolladores del AWS SDK y ejemplos de código, consulte Uso de este servicio con un SDK de AWS. En este tema también se incluye información sobre cómo comenzar a utilizar el SDK y detalles sobre sus versiones anteriores.

Uso de **`ListBucketAnalyticsConfigurations`** con un AWS SDK o la CLI

Los siguientes ejemplos de código muestran cómo utilizar `ListBucketAnalyticsConfigurations`.

CLI

AWS CLI

Recuperar una lista de configuraciones de análisis para un bucket

El siguiente `list-bucket-analytics-configurations` recupera una lista de configuraciones de análisis para el bucket especificado.

```
aws s3api list-bucket-analytics-configurations \
    --bucket my-bucket
```

Salida:

```
{
    "AnalyticsConfigurationList": [
        {
            "StorageClassAnalysis": {},
            "Id": "1"
        }
    ],
    "IsTruncated": false
}
```

- Para obtener información sobre la API, consulte ListBucketAnalyticsConfigurations en la Referencia de comandos de la AWS CLI.

PowerShell

Herramientas para PowerShell

Ejemplo 1: este comando devuelve las primeras 100 configuraciones de análisis del bucket de S3 indicado.

```
Get-S3BucketAnalyticsConfigurationList -BucketName 's3casetestbucket'
```

- Para obtener información sobre la API, consulte ListBucketAnalyticsConfigurations en la Referencia de Cmdlet de AWS Tools for PowerShell.

Para obtener una lista completa de las guías para desarrolladores del AWS SDK y ejemplos de código, consulte Uso de este servicio con un SDK de AWS. En este tema también se incluye información sobre cómo comenzar a utilizar el SDK y detalles sobre sus versiones anteriores.

Uso de **`ListBucketInventoryConfigurations`** con un AWS SDK o la CLI

Los siguientes ejemplos de código muestran cómo utilizar `ListBucketInventoryConfigurations`.

CLI

AWS CLI

Recuperar una lista de las configuraciones de inventario de un bucket

En el siguiente ejemplo de `list-bucket-inventory-configurations`, se enumeran las configuraciones de inventario del bucket especificado.

```
aws s3api list-bucket-inventory-configurations \
    --bucket my-bucket
```

Salida:

```
{
    "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
}
```

- Para obtener información sobre la API, consulte ListBucketInventoryConfigurations en la Referencia de comandos de la AWS CLI.

PowerShell

Herramientas para PowerShell

Ejemplo 1: este comando devuelve las primeras 100 configuraciones de inventario del bucket de S3 indicado.

```
Get-S3BucketInventoryConfigurationList -BucketName 's3testbucket'
```

- Para obtener información sobre la API, consulte ListBucketInventoryConfigurations en la Referencia de Cmdlet de AWS Tools for PowerShell.

Para obtener una lista completa de las guías para desarrolladores del AWS SDK y ejemplos de código, consulte Uso de este servicio con un SDK de AWS. En este tema también se incluye información sobre cómo comenzar a utilizar el SDK y detalles sobre sus versiones anteriores.

Uso de **`ListBuckets`** con un AWS SDK o la CLI

Los siguientes ejemplos de código muestran cómo utilizar `ListBuckets`.

.NET

AWS SDK for .NET

> **Note**
>
> Hay más en GitHub. Busque el ejemplo completo y aprenda a configurar y ejecutar en el Repositorio de ejemplos de código de 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);
        }
    }
}
```

- Para obtener información sobre la API, consulte ListBuckets en la Referencia de la API de AWS SDK for .NET.

C++

SDK para C++

> **Note**
>
> Hay más en GitHub. Busque el ejemplo completo y aprenda a configurar y ejecutar en el Repositorio de ejemplos de código de 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;
}
```

- Para obtener información sobre la API, consulte ListBuckets en la Referencia de la API de AWS SDK for C++.

CLI

AWS CLI

El siguiente comando usa el comando `list-buckets` para mostrar los nombres de todos los buckets de Amazon S3 (en todas las regiones):

```
aws s3api list-buckets --query "Buckets[].Name"
```

La opción de consultas filtra la salida de `list-buckets` únicamente a los nombres de los buckets.

Para obtener más información sobre los buckets, consulte Trabajo con buckets de Amazon S3 en la Guía para desarrolladores de Amazon S3.

- Para obtener información sobre la API, consulte ListBuckets en la Referencia de comandos de la AWS CLI.

Go

SDK para Go V2

> **Note**
>
> Hay más en GitHub. Busque el ejemplo completo y aprenda a configurar y ejecutar en el Repositorio de ejemplos de código de 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
}
```

- Para obtener información sobre la API, consulte ListBuckets en la Referencia de la API de AWS SDK for Go.

Java

SDK para Java 2.x

> **Note**
>
> Hay más en GitHub. Busque el ejemplo completo y aprenda a configurar y ejecutar en el Repositorio de ejemplos de código de 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());
        }
    }
}
```

- Para obtener información sobre la API, consulte ListBuckets en la Referencia de la API de AWS SDK for Java 2.x.

JavaScript

SDK para JavaScript (v3)

> **Note**
>
> Hay más información en GitHub. Busque el ejemplo completo y aprenda a configurar y ejecutar en el Repositorio de ejemplos de código de AWS.

Obtener una lista de los buckets.

```
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);
  }
};
```

- Para obtener información, consulte la Guía para desarrolladores de AWS SDK for JavaScript.
- Para obtener información sobre la API, consulte ListBuckets en la Referencia de la API de AWS SDK for JavaScript.

PowerShell

Herramientas para PowerShell

Ejemplo 1: este comando devuelve todos los buckets de S3.

```
Get-S3Bucket
```

Ejemplo 2: este comando devuelve un bucket denominado “test-files”

```
Get-S3Bucket -BucketName test-files
```

- Para obtener información sobre la API, consulte ListBuckets en la Referencia de Cmdlet de AWS Tools for PowerShell.

Python

SDK para Python (Boto3)

> **Note**
>
> Hay más en GitHub. Busque el ejemplo completo y aprenda a configurar y ejecutar en el Repositorio de ejemplos de código de 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
```

- Para obtener información sobre la API, consulte ListBuckets en la Referencia de la API de AWS SDK para Python (Boto3).

Ruby

SDK para Ruby

> **Note**
>
> Hay más en GitHub. Busque el ejemplo completo y aprenda a configurar y ejecutar en el Repositorio de ejemplos de código de 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__
```

- Para obtener información sobre la API, consulte ListBuckets en la Referencia de la API de AWS SDK for Ruby.

Rust

SDK para Rust

> **Note**
>
> Hay más información en GitHub. Busque el ejemplo completo y aprenda a configurar y ejecutar en el Repositorio de ejemplos de código de 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(())
}
```

- Para obtener información sobre la API, consulte ListBuckets en la Referencia de la API de AWS SDK para Rust.

Swift

SDK para Swift

> **Note**
>
> Esto es documentación preliminar para un SDK en versión preliminar. Está sujeta a cambios.

> **Note**
>
> Hay más en GitHub. Busque el ejemplo completo y aprenda a configurar y ejecutar en el Repositorio de ejemplos de código de 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
}
```

- Para obtener información acerca de la API, consulte ListBuckets en la Referencia de la API del AWS SDK para Swift.

Para obtener una lista completa de las guías para desarrolladores del AWS SDK y ejemplos de código, consulte Uso de este servicio con un SDK de AWS. En este tema también se incluye información sobre cómo comenzar a utilizar el SDK y detalles sobre sus versiones anteriores.

Uso de **`ListMultipartUploads`** con un AWS SDK o la CLI

Los siguientes ejemplos de código muestran cómo utilizar `ListMultipartUploads`.

Los ejemplos de acciones son extractos de código de programas más grandes y deben ejecutarse en contexto. Puede ver esta acción en contexto en el siguiente ejemplo de código:

- Eliminación de cargas multiparte incompletas

CLI

AWS CLI

El siguiente comando muestra todas las cargas multiparte activas de un bucket denominado `my-bucket`:

```
aws s3api list-multipart-uploads --bucket my-bucket
```

Salida:

```
{
    "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": []
}
```

Las cargas multiparte en curso conllevan costos de almacenamiento en Amazon S3. Complete o anule una carga multiparte activa para eliminar sus partes de su cuenta.

- Para obtener información sobre la API, consulte ListMultipartUploads en la Referencia de comandos de la AWS CLI.

Java

SDK para Java 2.x

> **Note**
>
> Hay más en GitHub. Busque el ejemplo completo y aprenda a configurar y ejecutar en el Repositorio de ejemplos de código de 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);
        }
    }
}
```

- Para obtener información sobre la API, consulte ListMultipartUploads en la Referencia de la API de AWS SDK for Java 2.x.

Para obtener una lista completa de las guías para desarrolladores del AWS SDK y ejemplos de código, consulte Uso de este servicio con un SDK de AWS. En este tema también se incluye información sobre cómo comenzar a utilizar el SDK y detalles sobre sus versiones anteriores.

Uso de **`ListObjectVersions`** con un AWS SDK o la CLI

Los siguientes ejemplos de código muestran cómo utilizar `ListObjectVersions`.

Los ejemplos de acciones son extractos de código de programas más grandes y deben ejecutarse en contexto. Puede ver esta acción en contexto en el siguiente ejemplo de código:

- Trabajo con objetos con control de versiones

.NET

AWS SDK for .NET

> **Note**
>
> Hay más información en GitHub. Busque el ejemplo completo y aprenda a configurar y ejecutar en el Repositorio de ejemplos de código de 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}'");
        }
    }
}
```

- Para obtener información sobre la API, consulte ListObjectVersions en la Referencia de la API de AWS SDK for .NET.

CLI

AWS CLI

El siguiente comando recupera la información de la versión de un objeto en un bucket denominado `my-bucket`:

```
aws s3api list-object-versions --bucket my-bucket --prefix index.html
```

Salida:

```
{
    "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
        }
    ]
}
```

- Para obtener información sobre la API, consulte ListObjectVersions en la Referencia de comandos de la AWS CLI.

Go

SDK para Go V2

> **Note**
>
> Hay más en GitHub. Busque el ejemplo completo y aprenda a configurar y ejecutar en el Repositorio de ejemplos de código de 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
}
```

- Para obtener información sobre la API, consulte ListObjectVersions en la Referencia de la API de AWS SDK for Go.

Rust

SDK para Rust

> **Note**
>
> Hay más información en GitHub. Busque el ejemplo completo y aprenda a configurar y ejecutar en el Repositorio de ejemplos de código de 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(())
}
```

- Para obtener información sobre la API, consulte ListObjectVersions en la Referencia de la API de AWS SDK para Rust.

Para obtener una lista completa de las guías para desarrolladores del AWS SDK y ejemplos de código, consulte Uso de este servicio con un SDK de AWS. En este tema también se incluye información sobre cómo comenzar a utilizar el SDK y detalles sobre sus versiones anteriores.

Uso de **`ListObjects`** con un AWS SDK o la CLI

Los siguientes ejemplos de código muestran cómo utilizar `ListObjects`.

Los ejemplos de acciones son extractos de código de programas más grandes y deben ejecutarse en contexto. Puede ver esta acción en contexto en el siguiente ejemplo de código:

- Crear una página web que enumere los objetos de Amazon S3

CLI

AWS CLI

En el siguiente ejemplo se utiliza el comando `list-objects` para mostrar los nombres de todos los objetos del bucket especificado:

```
aws s3api list-objects --bucket text-content --query 'Contents[].{Key: Key, Size:
 Size}'
```

En el ejemplo se utiliza el argumento `--query` para filtrar la salida de `list-objects` hasta el valor de la clave y el tamaño de cada objeto

Para obtener más información sobre los objetos, consulte Trabajo con objetos de Amazon S3 en la Guía para desarrolladores de Amazon S3.

- Para obtener detalles de la API, consulte ListObjects en la Referencia de comandos de AWS CLI.

PowerShell

Herramientas para PowerShell

Ejemplo 1: este comando recupera la información sobre todos los elementos del bucket “test-files”.

```
Get-S3Object -BucketName test-files
```

Ejemplo 2: este comando recupera la información sobre el elemento “sample.txt” del bucket “test-files”.

```
Get-S3Object -BucketName test-files -Key sample.txt
```

Ejemplo 3: este comando recupera la información sobre todos los elementos con el prefijo “sample” del bucket “test-files”.

```
Get-S3Object -BucketName test-files -KeyPrefix sample
```

- Para obtener información sobre la API, consulte ListObjects en la Referencia de Cmdlet de AWS Tools for PowerShell.

Para obtener una lista completa de las guías para desarrolladores del AWS SDK y ejemplos de código, consulte Uso de este servicio con un SDK de AWS. En este tema también se incluye información sobre cómo comenzar a utilizar el SDK y detalles sobre sus versiones anteriores.

Uso de **`ListObjectsV2`** con un AWS SDK o la CLI

Los siguientes ejemplos de código muestran cómo utilizar `ListObjectsV2`.

Los ejemplos de acciones son extractos de código de programas más grandes y deben ejecutarse en contexto. Puede ver esta acción en contexto en el siguiente ejemplo de código:

- Introducción a los buckets y objetos

.NET

AWS SDK for .NET

> **Note**
>
> Hay más información en GitHub. Busque el ejemplo completo y aprenda a configurar y ejecutar en el Repositorio de ejemplos de código de 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;
            }
        }
```

Muestre objetos con un paginador.

```
    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}");
                }
            }
        }
    }
```

- Para obtener información sobre la API, consulte ListObjectsV2 en la Referencia de la API de AWS SDK for .NET.

Bash

AWS CLI con Bash script

> **Note**
>
> Hay más información en GitHub. Busque el ejemplo completo y aprenda a configurar y ejecutar en el Repositorio de ejemplos de código de 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
}
```

- Para obtener información sobre la API, consulte ListObjectsV2 en la Referencia de comandos de AWS CLI.

C++

SDK para C++

> **Note**
>
> Hay más en GitHub. Busque el ejemplo completo y aprenda a configurar y ejecutar en el Repositorio de ejemplos de código de AWS.

```
bool AwsDoc::S3::listObjects(const Aws::String &bucketName,
                             Aws::Vector<Aws::String> &keysResult,
                             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;
        keysResult.push_back(object.GetKey());
    }

    return true;
}
```

- Para obtener información sobre la API, consulte ListObjectsV2 en la Referencia de la API de AWS SDK for C++.

CLI

AWS CLI

Obtención de una lista de objetos en un bucket

En el siguiente ejemplo de `list-objects-v2` se muestran los objetos del bucket especificado.

```
aws s3api list-objects-v2 \
    --bucket my-bucket
```

Salida:

```
{
    "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
        }
    ]
}
```

- Para obtener información sobre la API, consulte ListObjectsV2 en la Referencia de comandos de AWS CLI.

Go

SDK para Go V2

> **Note**
>
> Hay más en GitHub. Busque el ejemplo completo y aprenda a configurar y ejecutar en el Repositorio de ejemplos de código de 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
}
```

- Para obtener información sobre la API, consulte ListObjectsV2 en la Referencia de la API de AWS SDK for Go.

Java

SDK para Java 2.x

> **Note**
>
> Hay más en GitHub. Busque el ejemplo completo y aprenda a configurar y ejecutar en el Repositorio de ejemplos de código de 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;
    }
}
```

Muestre objetos mediante paginación.

```
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);
        }
    }
}
```

- Para obtener información sobre la API, consulte ListObjectsV2 en la Referencia de la API de AWS SDK for Java 2.x.

JavaScript

SDK para JavaScript (v3)

> **Note**
>
> Hay más información en GitHub. Busque el ejemplo completo y aprenda a configurar y ejecutar en el Repositorio de ejemplos de código de AWS.

Enumere todos los objetos del bucket. Si hay más de un objeto, se usarán isTruncated y NextContinuationToken para repetir la lista completa.

```
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);
  }
};
```

- Para obtener información sobre la API, consulte ListObjectsV2 en la Referencia de la API de AWS SDK for JavaScript.

Kotlin

SDK para Kotlin

> **Note**
>
> Hay más información en GitHub. Busque el ejemplo completo y aprenda a configurar y ejecutar en el Repositorio de ejemplos de código de 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
```

- Para obtener información sobre la API, consulte ListObjectsV2 en la Referencia de la API de AWS SDK para Kotlin.

PHP

SDK para PHP

> **Note**
>
> Hay más en GitHub. Busque el ejemplo completo y aprenda a configurar y ejecutar en el Repositorio de ejemplos de código de AWS.

Obtenga una lista de objetos de un bucket.

```
        $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.");
        }
```

- Para obtener información sobre la API, consulte ListObjectsV2 en la Referencia de la API de AWS SDK for PHP.

Python

SDK para Python (Boto3)

> **Note**
>
> Hay más en GitHub. Busque el ejemplo completo y aprenda a configurar y ejecutar en el Repositorio de ejemplos de código de 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
```

- Para obtener información sobre la API, consulte ListObjectsV2 en la Referencia de la API de AWS SDK para Python (Boto3).

Ruby

SDK para Ruby

> **Note**
>
> Hay más en GitHub. Busque el ejemplo completo y aprenda a configurar y ejecutar en el Repositorio de ejemplos de código de 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__
```

- Para obtener información sobre la API, consulte ListObjectsV2 en la Referencia de la API de AWS SDK for Ruby.

Rust

SDK para Rust

> **Note**
>
> Hay más información en GitHub. Busque el ejemplo completo y aprenda a configurar y ejecutar en el Repositorio de ejemplos de código de 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(())
}
```

- Para obtener información sobre la API, consulte ListObjectsV2 en la Referencia de la API de AWS SDK para Rust.

SAP ABAP

SDK de SAP ABAP

> **Note**
>
> Hay más en GitHub. Busque el ejemplo completo y aprenda a configurar y ejecutar en el Repositorio de ejemplos de código de 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.
```

- Para obtener información sobre la API, consulte ListObjectsV2 en la Referencia de la API de AWS SDK para SAP ABAP.

Swift

SDK para Swift

> **Note**
>
> Esto es documentación preliminar para un SDK en versión preliminar. Está sujeta a cambios.

> **Note**
>
> Hay más en GitHub. Busque el ejemplo completo y aprenda a configurar y ejecutar en el Repositorio de ejemplos de código de 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
    }
```

- Para obtener información acerca de la API, consulte ListObjectsV2 en la Referencia de la API del SDK de AWS para Swift.

Para obtener una lista completa de las guías para desarrolladores del AWS SDK y ejemplos de código, consulte Uso de este servicio con un SDK de AWS. En este tema también se incluye información sobre cómo comenzar a utilizar el SDK y detalles sobre sus versiones anteriores.

Uso de **PutBucketAccelerateConfiguration** con un AWS SDK o la CLI

Los siguientes ejemplos de código muestran cómo utilizar `PutBucketAccelerateConfiguration`.

.NET

AWS SDK for .NET

> **Note**
>
> Hay más en GitHub. Busque el ejemplo completo y aprenda a configurar y ejecutar en el Repositorio de ejemplos de código de 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");
            }
        }
    }
```

- Para obtener información sobre la API, consulte PutBucketAccelerateConfiguration en la Referencia de la API de AWS SDK for .NET.

CLI

AWS CLI

Establecer la configuración acelerada de un bucket

En el siguiente ejemplo de `put-bucket-accelerate-configuration`, se habilita la configuración acelerada para el bucket especificado.

```
aws s3api put-bucket-accelerate-configuration \
    --bucket my-bucket \
    --accelerate-configuration Status=Enabled
```

Este comando no genera ninguna salida.

- Para obtener información sobre la API, consulte PutBucketAccelerateConfiguration en Referencia de comandos de la AWS CLI.

PowerShell

Herramientas para PowerShell

Ejemplo 1: este comando habilita la aceleración de transferencia para el bucket de S3 indicado.

```
$statusVal = New-Object Amazon.S3.BucketAccelerateStatus('Enabled')
Write-S3BucketAccelerateConfiguration -BucketName 's3testbucket' -
AccelerateConfiguration_Status $statusVal
```

- Para obtener información sobre la API, consulte PutBucketAccelerateConfiguration en la Referencia de Cmdlet de AWS Tools for PowerShell.

Para obtener una lista completa de las guías para desarrolladores del AWS SDK y ejemplos de código, consulte Uso de este servicio con un SDK de AWS. En este tema también se incluye información sobre cómo comenzar a utilizar el SDK y detalles sobre sus versiones anteriores.

Uso de **`PutBucketAcl`** con un AWS SDK o la CLI

Los siguientes ejemplos de código muestran cómo utilizar `PutBucketAcl`.

Los ejemplos de acciones son extractos de código de programas más grandes y deben ejecutarse en contexto. Puede ver esta acción en contexto en el siguiente ejemplo de código:

- Administrar listas de control de acceso (ACL)

.NET

AWS SDK for .NET

> **Note**
>
> Hay más información en GitHub. Busque el ejemplo completo y aprenda a configurar y ejecutar en el Repositorio de ejemplos de código de 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;
        }
```

- Para obtener información sobre la API, consulte PutBucketAcl en la Referencia de la API de AWS SDK for .NET.

C++

SDK para C++

> **Note**
>
> Hay más en GitHub. Busque el ejemplo completo y aprenda a configurar y ejecutar en el Repositorio de ejemplos de código de 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;
}
```

- Para obtener información sobre la API, consulte PutBucketAcl en la Referencia de la API de AWS SDK for C++.

CLI

AWS CLI

Este ejemplo otorga `full control` a dos usuarios de AWS user1@example.com y user2@example.com) y permiso de `read` a todos los usuarios:

```
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
```

Consulte http://docs.aws.amazon.com/AmazonS3/latest/API/RESTBucketPUTacl.html para obtener más información sobre las ACL personalizadas (los comandos s3api de ACL, como `put-bucket-acl`, utilizan la misma notación abreviada para el argumentos).

- Para obtener información sobre la API, consulte PutBucketAcl en la Referencia de comandos de la AWS CLI.

Java

SDK para Java 2.x

> **Note**
>
> Hay más en GitHub. Busque el ejemplo completo y aprenda a configurar y ejecutar en el Repositorio de ejemplos de código de 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);
        }
    }
}
```

- Para obtener información sobre la API, consulte PutBucketAcl en la Referencia de la API de AWS SDK for Java 2.x.

JavaScript

SDK para JavaScript (v3)

> **Note**
>
> Hay más información en GitHub. Busque el ejemplo completo y aprenda a configurar y ejecutar en el Repositorio de ejemplos de código de AWS.

Coloque la ACL del bucket.

```
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);
  }
};
```

- Para obtener información, consulte la Guía para desarrolladores de AWS SDK for JavaScript.
- Para obtener información sobre la API, consulte PutBucketAcl en la Referencia de la API de AWS SDK for JavaScript.

Kotlin

SDK para Kotlin

> **Note**
>
> Hay más información en GitHub. Busque el ejemplo completo y aprenda a configurar y ejecutar en el Repositorio de ejemplos de código de 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")
    }
}
```

- Para obtener información sobre la API, consulte PutBucketAcl en la Referencia de la API de AWS SDK para Kotlin.

Python

SDK para Python (Boto3)

> **Note**
>
> Hay más en GitHub. Busque el ejemplo completo y aprenda a configurar y ejecutar en el Repositorio de ejemplos de código de 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
```

- Para obtener información sobre la API, consulte PutBucketAcl en la Referencia de la API de AWS SDK para Python (Boto3).

Para obtener una lista completa de las guías para desarrolladores del AWS SDK y ejemplos de código, consulte Uso de este servicio con un SDK de AWS. En este tema también se incluye información sobre cómo comenzar a utilizar el SDK y detalles sobre sus versiones anteriores.

Uso de **`PutBucketCors`** con un AWS SDK o la CLI

Los siguientes ejemplos de código muestran cómo utilizar `PutBucketCors`.

.NET

AWS SDK for .NET

> **Note**
>
> Hay más en GitHub. Busque el ejemplo completo y aprenda a configurar y ejecutar en el Repositorio de ejemplos de código de 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);
        }
```

- Para obtener información sobre la API, consulte PutBucketCors en la Referencia de la API de AWS SDK for .NET.

CLI

AWS CLI

El siguiente ejemplo habilita solicitudes PUT, POST y DELETE desde www.ejemplo.com, y habilita solicitudes GET desde cualquier dominio:

```
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
```

```
    }
  ]
}
```

- Para obtener información acerca de la API, consulte PutBucketCors en la Referencia de comandos de la AWS CLI.

Java

SDK para Java 2.x

> **Note**
>
> Hay más en GitHub. Busque el ejemplo completo y aprenda a configurar y ejecutar en el Repositorio de ejemplos de código de 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);
        }
    }
}
```

- Para obtener información sobre la API, consulte PutBucketCors en la Referencia de la API de AWS SDK for Java 2.x.

JavaScript

SDK para JavaScript (v3)

> **Note**
>
> Hay más información en GitHub. Busque el ejemplo completo y aprenda a configurar y ejecutar en el Repositorio de ejemplos de código de AWS.

Añada una regla 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);
  }
};
```

- Para obtener información, consulte la Guía para desarrolladores de AWS SDK for JavaScript.
- Para obtener información sobre la API, consulte PutBucketCors en la Referencia de la API de AWS SDK for JavaScript.

Python

SDK para Python (Boto3)

> **Note**
>
> Hay más en GitHub. Busque el ejemplo completo y aprenda a configurar y ejecutar en el Repositorio de ejemplos de código de 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
```

- Para obtener información sobre las API, consulte PutBucketCors en la Referencia de la API de AWS SDK para Python (Boto3).

Ruby

SDK para Ruby

> **Note**
>
> Hay más en GitHub. Busque el ejemplo completo y aprenda a configurar y ejecutar en el Repositorio de ejemplos de código de 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
```

- Para obtener información sobre la API, consulte PutBucketCors en la Referencia de la API de AWS SDK for Ruby.

Para obtener una lista completa de las guías para desarrolladores del AWS SDK y ejemplos de código, consulte Uso de este servicio con un SDK de AWS. En este tema también se incluye información sobre cómo comenzar a utilizar el SDK y detalles sobre sus versiones anteriores.

Uso de **PutBucketEncryption** con un AWS SDK o la CLI

Los siguientes ejemplos de código muestran cómo utilizar `PutBucketEncryption`.

CLI

AWS CLI

Configurar el cifrado del lado del servidor de un bucket

En el siguiente ejemplo de `put-bucket-encryption` se establece el cifrado AES256 como predeterminado para el bucket especificado.

```
aws s3api put-bucket-encryption \
    --bucket my-bucket \
    --server-side-encryption-configuration '{"Rules":
 [{"ApplyServerSideEncryptionByDefault": {"SSEAlgorithm": "AES256"}}]}'
```

Este comando no genera ninguna salida.

- Para obtener información sobre la API, consulte PutBucketEncryption en la Referencia de comandos de la AWS CLI.

PowerShell

Herramientas para PowerShell

Ejemplo 1: este comando habilita el cifrado AES256 predeterminado del servidor con claves administradas de Amazon S3 (SSE-S3) en el bucket indicado.

```
$Encryptionconfig = @{ServerSideEncryptionByDefault =
 @{ServerSideEncryptionAlgorithm = "AES256"}}
Set-S3BucketEncryption -BucketName 's3testbucket' -
ServerSideEncryptionConfiguration_ServerSideEncryptionRule $Encryptionconfig
```

- Para obtener información sobre la API, consulte PutBucketEncryption en la Referencia de Cmdlet de AWS Tools for PowerShell.

Para obtener una lista completa de las guías para desarrolladores del AWS SDK y ejemplos de código, consulte Uso de este servicio con un SDK de AWS. En este tema también se incluye información sobre cómo comenzar a utilizar el SDK y detalles sobre sus versiones anteriores.

Uso de **`PutBucketLifecycleConfiguration`** con un AWS SDK o la CLI

Los siguientes ejemplos de código muestran cómo utilizar `PutBucketLifecycleConfiguration`.

Los ejemplos de acciones son extractos de código de programas más grandes y deben ejecutarse en contexto. Puede ver esta acción en su contexto en los siguientes ejemplos de código:

- Eliminación de cargas multiparte incompletas
- Trabajo con objetos con control de versiones

.NET

AWS SDK for .NET

> **Note**
>
> Hay más información en GitHub. Busque el ejemplo completo y aprenda a configurar y ejecutar en el Repositorio de ejemplos de código de 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);
        }
```

- Para obtener información sobre la API, consulte PutBucketLifecycleConfiguration en la Referencia de la API de AWS SDK for .NET.

CLI

AWS CLI

El comando siguiente aplica una configuración del ciclo de vida a un bucket denominado `my-bucket`:

```
aws s3api put-bucket-lifecycle-configuration --bucket my-bucket --lifecycle-configuration  file://lifecycle.json
```

El archivo `lifecycle.json` es un documento JSON en la carpeta actual que especifica dos reglas:

```
{
    "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 primera regla mueve los archivos con el prefijo `rotated` a Glacier en la fecha especificada. La segunda regla mueve las versiones del objeto antiguas a Glacier cuando ya

no están actualizadas. Para obtener más información sobre los formatos de marca temporal permitidos, consulte Especificación de valores de parámetros para la Guía del usuario de la AWS CLI.

- Para obtener información sobre la API, consulte PutBucketLifecycleConfiguration en la Referencia de comandos de la AWS CLI.

Java

SDK para Java 2.x

> **Note**
>
> Hay más en GitHub. Busque el ejemplo completo y aprenda a configurar y ejecutar en el Repositorio de ejemplos de código de 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);
                }
        }
}
```

- Para obtener información sobre la API, consulte PutBucketLifecycleConfiguration en la Referencia de la API de AWS SDK for Java 2.x.

Python

SDK para Python (Boto3)

> **Note**
>
> Hay más en GitHub. Busque el ejemplo completo y aprenda a configurar y ejecutar en el Repositorio de ejemplos de código de 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
```

- Para obtener información sobre la API, consulte PutBucketLifecycleConfiguration en la Referencia de la API de AWS SDK para Python (Boto3).

Para obtener una lista completa de las guías para desarrolladores del AWS SDK y ejemplos de código, consulte Uso de este servicio con un SDK de AWS. En este tema también se incluye información sobre cómo comenzar a utilizar el SDK y detalles sobre sus versiones anteriores.

Uso de **PutBucketLogging** con un AWS SDK o la CLI

Los siguientes ejemplos de código muestran cómo utilizar `PutBucketLogging`.

.NET

AWS SDK for .NET

> **Note**
>
> Hay más en GitHub. Busque el ejemplo completo y aprenda a configurar y ejecutar en el Repositorio de ejemplos de código de 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();
        }
    }
```

- Para obtener información sobre la API, consulte PutBucketLogging en la Referencia de la API de AWS SDK for .NET.

CLI

AWS CLI

Ejemplo 1: Configuración del registro de políticas de bucket

El siguiente ejemplo de `put-bucket-logging` establece la política de registro para MyBucket. En primer lugar, conceda al servicio de registro el permiso de entidad principal en la política de bucket mediante el comando `put-bucket-policy`.

```
aws s3api put-bucket-policy \
    --bucket MyBucket \
    --policy file://policy.json
```

Contenidos 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"}
            }
        }
    ]
```

```
}
```

Para aplicar la política de registro, use `put-bucket-logging`.

```
aws s3api put-bucket-logging \
    --bucket MyBucket \
    --bucket-logging-status file://logging.json
```

Contenidos de `logging.json`:

```
{
     "LoggingEnabled": {
         "TargetBucket": "MyBucket",
         "TargetPrefix": "Logs/"
     }
 }
```

El comando `put-bucket-policy` es necesario para conceder permisos `s3:PutObject` a la entidad principal del servicio de registro.

Para obtener más información, consulte Registro de acceso al servidor de Amazon S3 en la Guía del usuario de Amazon S3.

Ejemplo 2: Establecimiento de una política de bucket para registrar el acceso a un solo usuario

El siguiente ejemplo de `put-bucket-logging` establece la política de registro para MyBucket. El usuario de AWS bob@example.com tendrá el control total sobre los archivos de registro y nadie más tendrá acceso a ellos. En primer lugar, conceda permiso de S3 mediante `put-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
```

A continuación, aplique la política de registro mediante `put-bucket-logging`.

```
aws s3api put-bucket-logging \
    --bucket MyBucket \
```

```
    --bucket-logging-status file://logging.json
```

Contenidos de `logging.json`:

```
{
    "LoggingEnabled": {
        "TargetBucket": "MyBucket",
        "TargetPrefix": "MyBucketLogs/",
        "TargetGrants": [
            {
                "Grantee": {
                    "Type": "AmazonCustomerByEmail",
                    "EmailAddress": "bob@example.com"
                },
                "Permission": "FULL_CONTROL"
            }
        ]
    }
}
```

el comando `put-bucket-acl` es necesario para conceder los permisos necesarios (write y read-acp) al sistema de entrega de registros de S3.

Para obtener más información, consulte Registro de acceso al servidor de Amazon S3 en la Guía para desarrolladores de Amazon S3.

- Para obtener información sobre la API, consulte PutBucketLogging en la Referencia de comandos de la AWS CLI.

Para obtener una lista completa de las guías para desarrolladores del AWS SDK y ejemplos de código, consulte Uso de este servicio con un SDK de AWS. En este tema también se incluye información sobre cómo comenzar a utilizar el SDK y detalles sobre sus versiones anteriores.

Uso de **`PutBucketNotification`** con un AWS SDK o la CLI

Los siguientes ejemplos de código muestran cómo utilizar `PutBucketNotification`.

CLI

AWS CLI

Aplica una configuración de notificación a un bucket denominado `my-bucket`:

```
aws s3api put-bucket-notification --bucket my-bucket --notification-
configuration file://notification.json
```

El archivo `notification.json` es un documento JSON en la carpeta actual que especifica un tema de SNS y un tipo de evento para supervisar:

```
{
  "TopicConfiguration": {
    "Event": "s3:ObjectCreated:*",
    "Topic": "arn:aws:sns:us-west-2:123456789012:s3-notification-topic"
  }
}
```

El tema de SNS debe tener una política de IAM adjunta que permita a Amazon S3 publicar en él:

```
{
 "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"
    }
   }
  }
 ]
}
```

- Para obtener información sobre la API, consulte PutBucketNotification en la Referencia de comandos de la AWS CLI.

PowerShell

Herramientas para PowerShell

Ejemplo 1: este ejemplo configura la configuración del tema de SNS para el evento ObjectRemovedDelete de S3 y se habilita la notificación para el bucket de S3 indicado

```
$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
```

Ejemplo 2: este ejemplo habilita las notificaciones de ObjectCreatedAll para el bucket indicado y las envía a la función de 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
```

Ejemplo 3: este ejemplo crea 2 configuraciones de Lambda diferentes sobre la base de un sufijo clave diferente y las configura en un solo comando.

```
#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
```

- Para obtener información sobre la API, consulte PutBucketNotification en la Referencia de Cmdlet de AWS Tools for PowerShell.

Para obtener una lista completa de las guías para desarrolladores del AWS SDK y ejemplos de código, consulte Uso de este servicio con un SDK de AWS. En este tema también se incluye información sobre cómo comenzar a utilizar el SDK y detalles sobre sus versiones anteriores.

Uso de **PutBucketNotificationConfiguration** con un AWS SDK o la CLI

Los siguientes ejemplos de código muestran cómo utilizar `PutBucketNotificationConfiguration`.

Los ejemplos de acciones son extractos de código de programas más grandes y deben ejecutarse en contexto. Puede ver esta acción en su contexto en los siguientes ejemplos de código:

- Procese notificaciones de eventos de S3
- Envío de notificaciones de eventos a EventBridge

.NET

AWS SDK for .NET

> **Note**
>
> Hay más información en GitHub. Busque el ejemplo completo y aprenda a configurar y ejecutar en el Repositorio de ejemplos de código de 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}");
            }
        }
    }
```

- Para obtener información sobre la API, consulte PutBucketNotificationConfiguration en Referencia de la API de AWS SDK for .NET.

CLI

AWS CLI

Habilitación de las notificaciones especificadas en un bucket

El siguiente ejemplo de `put-bucket-notification-configuration` se aplica una configuración de notificación a un bucket llamado `my-bucket`. El archivo `notification.json` es un documento JSON en la carpeta actual que especifica un tema de SNS y un tipo de evento para supervisar.

```
aws s3api put-bucket-notification-configuration \
    --bucket my-bucket \
    --notification-configuration file://notification.json
```

Contenidos de `notification.json`:

```
{
    "TopicConfigurations": [
        {
            "TopicArn": "arn:aws:sns:us-west-2:123456789012:s3-notification-
topic",
            "Events": [
                "s3:ObjectCreated:*"
            ]
        }
    ]
}
```

El tema de SNS debe tener una política de IAM adjunta que permita a Amazon S3 publicar en él.

```
{
    "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"
                }
            }
        }
    ]
}
```

- Para obtener información sobre la API, consulte PutBucketNotificationConfiguration en Referencia de comandos de la AWS CLI.

Para obtener una lista completa de las guías para desarrolladores del AWS SDK y ejemplos de código, consulte Uso de este servicio con un SDK de AWS. En este tema también se incluye información sobre cómo comenzar a utilizar el SDK y detalles sobre sus versiones anteriores.

Uso de **PutBucketPolicy** con un AWS SDK o la CLI

Los siguientes ejemplos de código muestran cómo utilizar `PutBucketPolicy`.

C++

SDK para C++

> **Note**
>
> Hay más en GitHub. Busque el ejemplo completo y aprenda a configurar y ejecutar en el Repositorio de ejemplos de código de 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"
            "}";
}
```

- Para obtener información acerca de la API, consulte PutBucketPolicy en la referencia de la API de AWS SDK for C++.

CLI

AWS CLI

Este ejemplo permite a todos los usuarios recuperar cualquier objeto de MyBucket excepto los de MySecretFolder. También concede un permiso `put` y `delete` al usuario raíz de la cuenta de AWS `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/*"
      }

  ]
}
```

- Para obtener información sobre la API, consulte PutBucketPolicy en la Referencia de comandos de la AWS CLI.

Java

SDK para Java 2.x

> **Note**
>
> Hay más en GitHub. Busque el ejemplo completo y aprenda a configurar y ejecutar en el Repositorio de ejemplos de código de 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();
    }
}
```

- Para obtener información acerca de la API, consulte PutBucketPolicy en la referencia de la API de AWS SDK for Java 2.x.

JavaScript

SDK para JavaScript (v3)

> **Note**
>
> Hay más información en GitHub. Busque el ejemplo completo y aprenda a configurar y ejecutar en el Repositorio de ejemplos de código de AWS.

Añada la política.

```
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);
  }
};
```

- Para obtener información, consulte la Guía para desarrolladores de AWS SDK for JavaScript.
- Para obtener información sobre la API, consulte PutBucketPolicy en la Referencia de la API de AWS SDK for JavaScript.

Python

SDK para Python (Boto3)

> **Note**
>
> Hay más en GitHub. Busque el ejemplo completo y aprenda a configurar y ejecutar en el Repositorio de ejemplos de código de 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
```

- Para obtener información sobre la API, consulte PutBucketPolicy en la Referencia de la API de AWS SDK para Python (Boto3).

Ruby

SDK para Ruby

> **Note**
>
> Hay más en GitHub. Busque el ejemplo completo y aprenda a configurar y ejecutar en el Repositorio de ejemplos de código de 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
```

- Para obtener información acerca de la API, consulte PutBucketPolicy en la referencia de la API de AWS SDK for Ruby.

Para obtener una lista completa de las guías para desarrolladores del AWS SDK y ejemplos de código, consulte Uso de este servicio con un SDK de AWS. En este tema también se incluye información sobre cómo comenzar a utilizar el SDK y detalles sobre sus versiones anteriores.

Uso de **PutBucketReplication** con un AWS SDK o la CLI

Los siguientes ejemplos de código muestran cómo utilizar `PutBucketReplication`.

CLI

AWS CLI

Configurar la replicación de un bucket de S3

El siguiente ejemplo de `put-bucket-replication` aplica una configuración de replicación al bucket de S3 especificado.

```
aws s3api put-bucket-replication \
    --bucket AWSDOC-EXAMPLE-BUCKET1 \
    --replication-configuration file://replication.json
```

Contenidos 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"
            }
        }
    ]
}
```

El bucket de destino debe tener habilitado el control de versiones. El rol especificado debe tener permiso para escribir en el bucket de destino y tener una relación de confianza que permita a Amazon S3 asumir el rol.

Ejemplo de política de permisos de roles:

```
{
    "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/*"
        }
    ]
}
```

Ejemplo de política de relación de confianza:

```
{
    "Version": "2012-10-17",
    "Statement": [
```

```
        {
            "Effect": "Allow",
            "Principal": {
                "Service": "s3.amazonaws.com"
            },
            "Action": "sts:AssumeRole"
        }
    ]
}
```

Este comando no genera ninguna salida.

Para obtener más información, consulte Este es el título del tema en la guía del usuario de la consola de Amazon Simple Storage Service:

- Para obtener información sobre la API, consulte PutBucketReplication en la Referencia de comandos de la AWS CLI.

PowerShell

Herramientas para PowerShell

Ejemplo 1: este ejemplo establece una configuración de replicación con una sola regla que permite replicar en el bucket “exampletargetbucket” cualquier objeto nuevo creado con el prefijo de nombre de clave “TaxDocs” en el bucket “examplebucket”.

```
$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
```

Ejemplo 2: este ejemplo establece una configuración de replicación con varias reglas que permiten replicar en el bucket “exampletargetbucket” cualquier objeto nuevo creado con el prefijo de nombre de clave “TaxDocs” u “OtherDocs”. Los prefijos de claves no deben superponerse.

```
$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
```

Ejemplo 3: este ejemplo actualiza la configuración de replicación en el bucket especificado para inhabilitar la regla que controla la replicación de objetos con el prefijo de nombre de clave “TaxDocs” en el bucket “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
```

- Para obtener información sobre la API, consulte PutBucketReplication en la Referencia de Cmdlet de AWS Tools for PowerShell.

Para obtener una lista completa de las guías para desarrolladores del AWS SDK y ejemplos de código, consulte Uso de este servicio con un SDK de AWS. En este tema también se incluye información sobre cómo comenzar a utilizar el SDK y detalles sobre sus versiones anteriores.

Uso de **PutBucketRequestPayment** con un AWS SDK o la CLI

Los siguientes ejemplos de código muestran cómo utilizar `PutBucketRequestPayment`.

CLI

AWS CLI

Ejemplo 1: habilitar la configuración de “el solicitante paga” para un bucket

El siguiente ejemplo de `put-bucket-request-payment` habilita `requester pays` para el bucket especificado.

```
aws s3api put-bucket-request-payment \
    --bucket my-bucket \
    --request-payment-configuration '{"Payer":"Requester"}'
```

Este comando no genera ninguna salida.

Ejemplo 2: deshabilitar la configuración de “el solicitante paga” para un bucket

El siguiente ejemplo de `put-bucket-request-payment` deshabilita `requester pays` para el bucket especificado.

```
aws s3api put-bucket-request-payment \
    --bucket my-bucket \
    --request-payment-configuration '{"Payer":"BucketOwner"}'
```

Este comando no genera ninguna salida.

- Para obtener información sobre la API, consulte PutBucketRequestPayment en la Referencia de comandos de la AWS CLI.

PowerShell

Herramientas para PowerShell

Ejemplo 1: actualiza la configuración de pago de solicitud del bucket denominado “mybucket”, de modo que se cobre la descarga a la persona que solicita las descargas del bucket. De forma predeterminada, el propietario del bucket paga las descargas. Para volver a establecer el pago de la solicitud al modo predeterminado, use “BucketOwner” para el parámetro RequestPaymentConfiguration_Payer.

```
Write-S3BucketRequestPayment -BucketName mybucket -
RequestPaymentConfiguration_Payer Requester
```

- Para obtener información sobre la API, consulte PutBucketRequestPayment en la Referencia de Cmdlet de AWS Tools for PowerShell.

Para obtener una lista completa de las guías para desarrolladores del AWS SDK y ejemplos de código, consulte Uso de este servicio con un SDK de AWS. En este tema también se incluye información sobre cómo comenzar a utilizar el SDK y detalles sobre sus versiones anteriores.

Uso de **PutBucketTagging** con un AWS SDK o la CLI

Los siguientes ejemplos de código muestran cómo utilizar `PutBucketTagging`.

CLI

AWS CLI

El siguiente comando aplica una configuración de etiquetado a un bucket denominado `my-bucket`:

```
aws s3api put-bucket-tagging --bucket my-bucket --tagging file://tagging.json
```

El archivo `tagging.json` es un documento JSON en la carpeta actual que especifica etiquetas:

```
{
   "TagSet": [
     {
       "Key": "organization",
       "Value": "marketing"
     }
   ]
}
```

O aplique una configuración de etiquetado a `my-bucket` directamente desde la línea de comandos:

```
aws s3api put-bucket-tagging --bucket my-bucket --tagging
 'TagSet=[{Key=organization,Value=marketing}]'
```

- Para obtener información sobre la API, consulte PutBucketTagging en la Referencia de comandos de la AWS CLI.

PowerShell

Herramientas para PowerShell

Ejemplo 1: este comando aplica dos etiquetas a un bucket denominado **`cloudtrail-test-2018`**, una etiqueta con una clave de Stage y un valor de Test, y una etiqueta con una clave de Environment y un valor de Alpha. Para comprobar que las etiquetas se han añadido al bucket, ejecute **`Get-S3BucketTagging -BucketName bucket_name`**. Los resultados deben mostrar las etiquetas que ha aplicado al bucket en el primer comando. Tenga en cuenta que **`Write-S3BucketTagging`** sobrescribe todo el conjunto de etiquetas existente en un bucket. Para añadir o eliminar etiquetas individuales, ejecute los cmdlets Resource Groups y Tagging API, **`Add-RGTResourceTag`** y **`Remove-RGTResourceTag`**. Como alternativa, puede utilizar el editor de etiquetas de la consola de administración de AWS para administrar las etiquetas del bucket de S3.

```
Write-S3BucketTagging -BucketName cloudtrail-test-2018 -TagSet @( @{ Key="Stage";
 Value="Test" }, @{ Key="Environment"; Value="Alpha" } )
```

Ejemplo 2: este comando canaliza un bucket denominado **`cloudtrail-test-2018`** al cmdlet de **`Write-S3BucketTagging`**. Aplica las etiquetas Stage:Production y

Department:Finance al bucket. Tenga en cuenta que **`Write-S3BucketTagging`** sobrescribe todo el conjunto de etiquetas existente en un bucket.

```
Get-S3Bucket -BucketName cloudtrail-test-2018 | Write-S3BucketTagging
 -TagSet @( @{ Key="Stage"; Value="Production" }, @{ Key="Department";
 Value="Finance" } )
```

- Para obtener información sobre la API, consulte PutBucketTagging en la Referencia de Cmdlet de AWS Tools for PowerShell.

Para obtener una lista completa de las guías para desarrolladores del AWS SDK y ejemplos de código, consulte Uso de este servicio con un SDK de AWS. En este tema también se incluye información sobre cómo comenzar a utilizar el SDK y detalles sobre sus versiones anteriores.

Uso de **`PutBucketVersioning`** con un AWS SDK o la CLI

Los siguientes ejemplos de código muestran cómo utilizar `PutBucketVersioning`.

CLI

AWS CLI

El siguiente comando habilita el control de versiones en un bucket denominado `my-bucket`:

```
aws s3api put-bucket-versioning --bucket my-bucket --versioning-
configuration Status=Enabled
```

El siguiente comando habilita el control de versiones y usa un código mfa

```
aws s3api put-bucket-versioning --bucket my-bucket --versioning-
configuration Status=Enabled --mfa "SERIAL 123456"
```

- Para obtener información sobre la API, consulte PutBucketVersioning en la Referencia de comandos de la AWS CLI.

PowerShell

Herramientas para PowerShell

Ejemplo 1: el comando habilita el control de versiones para el bucket de S3 indicado.

```
Write-S3BucketVersioning -BucketName 's3testbucket' -VersioningConfig_Status
 Enabled
```

- Para obtener información sobre la API, consulte PutBucketVersioning en la Referencia de Cmdlet de AWS Tools for PowerShell.

Para obtener una lista completa de las guías para desarrolladores del AWS SDK y ejemplos de código, consulte Uso de este servicio con un SDK de AWS. En este tema también se incluye información sobre cómo comenzar a utilizar el SDK y detalles sobre sus versiones anteriores.

Uso de **PutBucketWebsite** con un AWS SDK o la CLI

Los siguientes ejemplos de código muestran cómo utilizar `PutBucketWebsite`.

.NET

AWS SDK for .NET

> **Note**
>
> Hay más en GitHub. Busque el ejemplo completo y aprenda a configurar y ejecutar en el Repositorio de ejemplos de código de AWS.

```
                // Put the website configuration.
                PutBucketWebsiteRequest putRequest = new
PutBucketWebsiteRequest()
                {
                    BucketName = bucketName,
                    WebsiteConfiguration = new WebsiteConfiguration()
                    {
                        IndexDocumentSuffix = indexDocumentSuffix,
                        ErrorDocument = errorDocument,
                    },
                };
                PutBucketWebsiteResponse response = await
client.PutBucketWebsiteAsync(putRequest);
```

- Para obtener información sobre la API, consulte PutBucketWebsite en la Referencia de la API de AWS SDK for .NET.

C++

SDK para C++

> **Note**
>
> Hay más en GitHub. Busque el ejemplo completo y aprenda a configurar y ejecutar en el Repositorio de ejemplos de código de 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();
}
```

- Para obtener información sobre la API, consulte PutBucketWebsite en la Referencia de la API de AWS SDK for C++.

CLI

AWS CLI

Aplica una configuración de sitio web estática a un bucket llamado `my-bucket`:

```
aws s3api put-bucket-website --bucket my-bucket --website-configuration file://
website.json
```

El archivo `website.json` es un documento JSON en la carpeta actual que especifica las páginas de índice y error del sitio web:

```
{
    "IndexDocument": {
        "Suffix": "index.html"
    },
    "ErrorDocument": {
        "Key": "error.html"
    }
}
```

- Para obtener información sobre la API, consulte PutBucketWebsite en la Referencia de comandos de la AWS CLI.

Java

SDK para Java 2.x

> **Note**
>
> Hay más en GitHub. Busque el ejemplo completo y aprenda a configurar y ejecutar en el Repositorio de ejemplos de código de 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);
        }
    }
}
```

- Para obtener información sobre la API, consulte PutBucketWebsite en la Referencia de la API de AWS SDK for Java 2.x.

JavaScript

SDK para JavaScript (v3)

> **Note**
>
> Hay más información en GitHub. Busque el ejemplo completo y aprenda a configurar y ejecutar en el Repositorio de ejemplos de código de AWS.

Establezca la configuración de sitio 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);
  }
};
```

- Para obtener información, consulte la Guía para desarrolladores de AWS SDK for JavaScript.
- Para obtener información sobre la API, consulte PutBucketWebsite en la Referencia de la API de AWS SDK for JavaScript.

PowerShell

Herramientas para PowerShell

Ejemplo 1: el comando habilita el alojamiento de sitios web para el bucket indicado con el documento de índice como “index.html” y el documento de error como “error.html”.

```
Write-S3BucketWebsite -BucketName 's3testbucket' -
WebsiteConfiguration_IndexDocumentSuffix 'index.html' -
WebsiteConfiguration_ErrorDocument 'error.html'
```

- Para obtener información sobre la API, consulte PutBucketWebsite en la Referencia de Cmdlet de AWS Tools for PowerShell.

Ruby

SDK para Ruby

> **Note**
>
> Hay más en GitHub. Busque el ejemplo completo y aprenda a configurar y ejecutar en el Repositorio de ejemplos de código de 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__
```

- Para obtener información sobre la API, consulte PutBucketWebsite en la Referencia de la API de AWS SDK for Ruby.

Para obtener una lista completa de las guías para desarrolladores del AWS SDK y ejemplos de código, consulte Uso de este servicio con un SDK de AWS. En este tema también se incluye información sobre cómo comenzar a utilizar el SDK y detalles sobre sus versiones anteriores.

Uso de **`PutObject`** con un AWS SDK o la CLI

Los siguientes ejemplos de código muestran cómo utilizar `PutObject`.

Los ejemplos de acciones son extractos de código de programas más grandes y deben ejecutarse en contexto. Puede ver esta acción en su contexto en los siguientes ejemplos de código:

- Introducción a los buckets y objetos
- Realización de un seguimiento de cargas y descargas
- Trabajo con la integridad de los objetos de Amazon S3

.NET

AWS SDK for .NET

> **Note**
>
> Hay más información en GitHub. Busque el ejemplo completo y aprenda a configurar y ejecutar en el Repositorio de ejemplos de código de 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;
            }
        }
```

Cargar un objeto con cifrado del lado del servidor.

```
    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");
            }
        }
    }
```

- Para obtener información sobre la API, consulte PutObject en la Referencia de la API de AWS SDK for .NET.

Bash

AWS CLI con script Bash

> **Note**
>
> Hay más información en GitHub. Busque el ejemplo completo y aprenda a configurar y ejecutar en el Repositorio de ejemplos de código de 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
}
```

- Para obtener información sobre la API, consulte PutObject en la Referencia de comandos de AWS CLI.

C++

SDK para C++

> **Note**
>
> Hay más en GitHub. Busque el ejemplo completo y aprenda a configurar y ejecutar en el Repositorio de ejemplos de código de 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();
}
```

- Para obtener información sobre la API, consulte PutObject en la Referencia de la API de AWS SDK for C++.

CLI

AWS CLI

En el siguiente ejemplo se utiliza el comando `put-object` para cargar un objeto a Amazon S3:

```
aws s3api put-object --bucket text-content --key dir-1/my_images.tar.bz2 --
body my_images.tar.bz2
```

En el siguiente ejemplo se muestra la carga de un archivo de vídeo (el archivo de vídeo se especifica mediante la sintaxis del sistema de archivos de 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
```

Para obtener más información acerca de la carga de objetos, consulte Carga de objetos en la Guía para desarrolladores de Amazon S3.

- Para obtener detalles de la API, consulte PutObject en la Referencia de comandos de AWS CLI.

Go

SDK para Go V2

> **Note**
>
> Hay más en GitHub. Busque el ejemplo completo y aprenda a configurar y ejecutar en el Repositorio de ejemplos de código de AWS.

Coloque un objeto en un bucket con la API de bajo nivel.

```
// 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
}
```

Cargue un objeto en un bucket con un administrador de transferencias.

```
// 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
}
```

- Para obtener información sobre la API, consulte PutObject en la Referencia de la API de AWS SDK for Go.

Java

SDK para Java 2.x

> **Note**
>
> Hay más en GitHub. Busque el ejemplo completo y aprenda a configurar y ejecutar en el Repositorio de ejemplos de código de AWS.

Cargue un archivo en un bucket con 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);
        }
    }
}
```

Utilice un S3TransferManager para cargar un archivo a un bucket. Vea el archivo completo y pruébelo.

```
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();
    }
```

Cargue un objeto en un bucket y configure las etiquetas con 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;
    }
}
```

Cargue un objeto en un bucket y configure los metadatos con 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);
        }
    }
}
```

Cargue un objeto en un bucket y configure un valor de retención de objetos con 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);
        }
    }
}
```

- Para obtener información sobre la API, consulte PutObject en la Referencia de la API de AWS SDK for Java 2.x.

JavaScript

SDK para JavaScript (v3)

> **Note**
>
> Hay más información en GitHub. Busque el ejemplo completo y aprenda a configurar y ejecutar en el Repositorio de ejemplos de código de AWS.

Cargue el objeto.

```
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);
  }
};
```

- Para obtener información, consulte la Guía para desarrolladores de AWS SDK for JavaScript.
- Para obtener información sobre la API, consulte PutObject en la Referencia de la API de AWS SDK for JavaScript.

Kotlin

SDK para Kotlin

> **Note**
>
> Hay más información en GitHub. Busque el ejemplo completo y aprenda a configurar y ejecutar en el Repositorio de ejemplos de código de 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}")
    }
}
```

- Para obtener información sobre la API, consulte PutObject en la Referencia de la API de AWS SDK para Kotlin.

PHP

SDK para PHP

> **Note**
>
> Hay más en GitHub. Busque el ejemplo completo y aprenda a configurar y ejecutar en el Repositorio de ejemplos de código de AWS.

Cargue un objeto en un bucket.

```
        $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.");
        }
```

- Para obtener información sobre la API, consulte PutObject en la Referencia de la API de AWS SDK for PHP.

PowerShell

Herramientas para PowerShell

Ejemplo 1: este comando carga el archivo único “local-sample.txt” a Amazon S3 y crea un objeto con la clave “sample.txt” en el bucket “test-files”.

```
Write-S3Object -BucketName test-files -Key "sample.txt" -File .\local-sample.txt
```

Ejemplo 2: este comando carga el archivo único “sample.txt” a Amazon S3 y crea un objeto con la clave “sample.txt” en el bucket “test-files”. Si no se proporciona el parámetro -Key, el nombre del archivo se utiliza como clave de objeto de S3.

```
Write-S3Object -BucketName test-files -File .\sample.txt
```

Ejemplo 3: este comando carga el archivo único “local-sample.txt” a Amazon S3 y crea un objeto con la clave “prefix/to/sample.txt” en el bucket “test-files”.

```
Write-S3Object -BucketName test-files -Key "prefix/to/sample.txt" -File .\local-
sample.txt
```

Ejemplo 4: este comando carga todos los archivos del subdirectorio “Scripts” al bucket “test-files” y aplica el prefijo de clave común “SampleScripts” a cada objeto. Cada archivo cargado tendrá una clave de “SampleScripts/filename”, donde “filename” varia.

```
Write-S3Object -BucketName test-files -Folder .\Scripts -KeyPrefix SampleScripts\
```

Ejemplo 5: este comando carga todos los archivos *.ps1 en el director local “Scripts” al bucket “test-files” y aplica el prefijo de clave común “SampleScripts” a cada objeto. Cada archivo cargado tendrá una clave de “SampleScripts/filename.ps1”, donde “filename” varia.

```
Write-S3Object -BucketName test-files -Folder .\Scripts -KeyPrefix SampleScripts\
 -SearchPattern *.ps1
```

Ejemplo 6: este comando crea un nuevo objeto S3 que contiene la cadena de contenido especificada con la clave “sample.txt”.

```
Write-S3Object -BucketName test-files -Key "sample.txt" -Content "object
 contents"
```

Ejemplo 7: este comando carga el archivo especificado (el nombre del archivo se usa como clave) y aplica las etiquetas especificadas al nuevo objeto.

```
Write-S3Object -BucketName test-files -File "sample.txt" -TagSet
 @{Key="key1";Value="value1"},@{Key="key2";Value="value2"}
```

Ejemplo 8: este comando carga de forma recursiva la carpeta especificada y aplica las etiquetas especificadas a todos los objetos nuevos.

```
Write-S3Object -BucketName test-files -Folder . -KeyPrefix "TaggedFiles" -Recurse
 -TagSet @{Key="key1";Value="value1"},@{Key="key2";Value="value2"}
```

- Para obtener información sobre la API, consulte PutObject en la Referencia de Cmdlet de AWS Tools for PowerShell.

Python

SDK para Python (Boto3)

> **Note**
>
> Hay más en GitHub. Busque el ejemplo completo y aprenda a configurar y ejecutar en el Repositorio de ejemplos de código de 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()
```

- Para obtener información sobre la API, consulte PutObject en la Referencia de la API de AWS SDK para Python (Boto3).

Ruby

SDK para Ruby

> **Note**
>
> Hay más en GitHub. Busque el ejemplo completo y aprenda a configurar y ejecutar en el Repositorio de ejemplos de código de AWS.

Cargue un archivo con un cargador administrado (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__
```

Cargue un archivo con 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__
```

Cargue un archivo con Object.put y añada cifrado del lado del servidor.

```
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__
```

- Para obtener información sobre la API, consulte PutObject en la Referencia de la API de AWS SDK for Ruby.

Rust

SDK para Rust

> **Note**
>
> Hay más información en GitHub. Busque el ejemplo completo y aprenda a configurar y ejecutar en el Repositorio de ejemplos de código de 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
}
```

- Para obtener información sobre la API, consulte PutObject en la Referencia de la API de AWS SDK para Rust.

SAP ABAP

SDK de SAP ABAP

> **Note**
>
> Hay más en GitHub. Busque el ejemplo completo y aprenda a configurar y ejecutar en el Repositorio de ejemplos de código de 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.
```

- Para obtener información sobre la API, consulte PutObject en la Referencia de la API de AWS SDK para SAP ABAP.

Swift

SDK para Swift

> **Note**
>
> Esto es documentación preliminar para un SDK en versión preliminar. Está sujeta a cambios.

> **Note**
>
> Hay más en GitHub. Busque el ejemplo completo y aprenda a configurar y ejecutar en el Repositorio de ejemplos de código de AWS.

Cargue un archivo del almacenamiento local en un bucket.

```
    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.data(fileData)

        let input = PutObjectInput(
            body: dataStream,
            bucket: bucket,
            key: key
        )
        _ = try await client.putObject(input: input)
    }
```

Cargue el contenido de un objeto de datos de Swift en un bucket.

```
    public func createFile(bucket: String, key: String, withData data: Data)
async throws {
```

```
        let dataStream = ByteStream.data(data)

        let input = PutObjectInput(
            body: dataStream,
            bucket: bucket,
            key: key
        )
        _ = try await client.putObject(input: input)
    }
```

- Para obtener información acerca de la API, consulte PutObject en la Referencia de la API del SDK de AWS para Swift.

Para obtener una lista completa de las guías para desarrolladores del AWS SDK y ejemplos de código, consulte Uso de este servicio con un SDK de AWS. En este tema también se incluye información sobre cómo comenzar a utilizar el SDK y detalles sobre sus versiones anteriores.

Uso de **`PutObjectAcl`** con un AWS SDK o la CLI

Los siguientes ejemplos de código muestran cómo utilizar `PutObjectAcl`.

Los ejemplos de acciones son extractos de código de programas más grandes y deben ejecutarse en contexto. Puede ver esta acción en su contexto en el siguiente ejemplo de código:

- Administrar listas de control de acceso (ACL)

C++

SDK para C++

> **Note**
>
> Hay más en GitHub. Busque el ejemplo completo y aprenda a configurar y ejecutar en el Repositorio de ejemplos de código de 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;
}
```

- Para obtener información sobre la API, consulte PutObjectAcl en la Referencia de la API de AWS SDK for C++.

CLI

AWS CLI

El siguiente ejemplo otorga `full control` a dos usuarios de AWS user1@example.com y user2@example.com) y permiso de `read` a todos los usuarios:

```
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
```

Consulte http://docs.aws.amazon.com/AmazonS3/latest/API/RESTBucketPUTacl.html para obtener más información sobre las ACL personalizadas (los comandos s3api de ACL, como `put-object-acl`, utilizan la misma notación abreviada para el argumentos).

- Para obtener información acerca de la API, consulte PutObjectAcl en la Referencia de comandos de la AWS CLI.

Python

SDK para Python (Boto3)

> **Note**
>
> Hay más en GitHub. Busque el ejemplo completo y aprenda a configurar y ejecutar en el Repositorio de ejemplos de código de 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
```

- Para obtener información sobre la API, consulte PutObjectAcl en la Referencia de la API de AWS SDK para Python (Boto3).

Para obtener una lista completa de las guías para desarrolladores del AWS SDK y ejemplos de código, consulte Uso de este servicio con un SDK de AWS. En este tema también se incluye información sobre cómo comenzar a utilizar el SDK y detalles sobre sus versiones anteriores.

Uso de **PutObjectLegalHold** con un AWS SDK o la CLI

Los siguientes ejemplos de código muestran cómo utilizar `PutObjectLegalHold`.

Los ejemplos de acciones son extractos de código de programas más grandes y deben ejecutarse en contexto. Puede ver esta acción en contexto en el siguiente ejemplo de código:

- Bloqueo de objetos de Amazon S3

.NET

AWS SDK for .NET

> **Note**
>
> Hay más información en GitHub. Busque el ejemplo completo y aprenda a configurar y ejecutar en el Repositorio de ejemplos de código de 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;
        }
    }
```

- Para obtener información sobre la API, consulte PutObjectLegalHold en la Referencia de la API de AWS SDK for .NET.

CLI

AWS CLI

Para aplicar una retención legal a un objeto

En el siguiente ejemplo de `put-object-legal-hold`, se establece una retención legal sobre el objeto `doc1.rtf`.

```
aws s3api put-object-legal-hold \
    --bucket my-bucket-with-object-lock \
    --key doc1.rtf \
    --legal-hold Status=ON
```

Este comando no genera ninguna salida.

- Para obtener información sobre la API, consulte PutObjectLegalHold en la Referencia de comandos de la AWS CLI.

Go

SDK para Go V2

> **Note**
>
> Hay más en GitHub. Busque el ejemplo completo y aprenda a configurar y ejecutar en el Repositorio de ejemplos de código de 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
}
```

- Para obtener información sobre la API, consulte PutObjectLegalHold en la Referencia de la API de AWS SDK for Go.

Java

SDK para Java 2.x

> **Note**
>
> Hay más en GitHub. Busque el ejemplo completo y aprenda a configurar y ejecutar en el Repositorio de ejemplos de código de 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 +".");
    }
```

- Para obtener información sobre la API, consulte PutObjectLegalHold en la Referencia de la API de AWS SDK for Java 2.x.

JavaScript

SDK para JavaScript (v3)

> **Note**
>
> Hay más información en GitHub. Busque el ejemplo completo y aprenda a configurar y ejecutar en el Repositorio de ejemplos de código de 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");
}
```

- Para obtener información sobre la API, consulte PutObjectLegalHold en la Referencia de la API de AWS SDK for JavaScript.

Python

SDK para Python (Boto3)

> **Note**
>
> Hay más en GitHub. Busque el ejemplo completo y aprenda a configurar y ejecutar en el Repositorio de ejemplos de código de AWS.

Ponga una retención legal en un objeto.

```
def set_legal_hold(s3_client, bucket: str, key: str) -> None:
    """
    Set a legal hold on a specific file in a bucket.

    Args:
        s3_client: Boto3 S3 client.
```

```
        bucket: The name of the bucket containing the file.
        key: The key of the file to set the legal hold on.
    """
    print()
    logger.info("Setting legal hold on file [%s] in bucket [%s]", key, bucket)
    try:
        before_status = "OFF"
        after_status = "ON"
        s3_client.put_object_legal_hold(
            Bucket=bucket, Key=key, LegalHold={"Status": after_status}
        )
        logger.debug(
            "Legal hold set successfully on file [%s] in bucket [%s]", key,
bucket
        )
        _print_legal_hold_update(bucket, key, before_status, after_status)
    except Exception as e:
        logger.error(
            "Failed to set legal hold on file [%s] in bucket [%s]: %s", key,
bucket, e
        )
```

- Para obtener información sobre la API, consulte PutObjectLegalHold en la Referencia de la API de AWS SDK para Python (Boto3).

Para obtener una lista completa de las guías para desarrolladores del AWS SDK y ejemplos de código, consulte Uso de este servicio con un SDK de AWS. En este tema también se incluye información sobre cómo comenzar a utilizar el SDK y detalles sobre sus versiones anteriores.

Uso de **`PutObjectLockConfiguration`** con un AWS SDK o la CLI

Los siguientes ejemplos de código muestran cómo utilizar `PutObjectLockConfiguration`.

Los ejemplos de acciones son extractos de código de programas más grandes y deben ejecutarse en contexto. Puede ver esta acción en contexto en el siguiente ejemplo de código:

- Bloqueo de objetos de Amazon S3

.NET

AWS SDK for .NET

> **Note**
>
> Hay más información en GitHub. Busque el ejemplo completo y aprenda a configurar y ejecutar en el Repositorio de ejemplos de código de AWS.

Establecimiento de la configuración de bloqueo de objetos de 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;
        }
    }
```

Establecimiento del período de retención predeterminado de 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;
        }
    }
```

- Para obtener información sobre la API, consulte PutObjectLockConfiguration en la Referencia de la API de AWS SDK for .NET.

CLI

AWS CLI

Para establecer la configuración de bloqueo de objetos en un bucket

En el siguiente ejemplo de `put-object-lock-configuration`, se establece un bloqueo de objetos de 50 días en el bucket especificado.

```
aws s3api put-object-lock-configuration \
    --bucket my-bucket-with-object-lock \
    --object-lock-configuration '{ "ObjectLockEnabled": "Enabled", "Rule":
 { "DefaultRetention": { "Mode": "COMPLIANCE", "Days": 50 }}}'
```

Este comando no genera ninguna salida.

- Para obtener información sobre la API, consulte PutObjectLockConfiguration en la Referencia de comandos de la AWS CLI.

Go

SDK para Go V2

> **Note**
>
> Hay más en GitHub. Busque el ejemplo completo y aprenda a configurar y ejecutar en el Repositorio de ejemplos de código de AWS.

Establecimiento de la configuración de bloqueo de objetos de 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
}
```

Establecimiento del período de retención predeterminado de 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
}
```

- Para obtener información sobre la API, consulte PutObjectLockConfiguration en la Referencia de la API de AWS SDK for Go.

Java

SDK para Java 2.x

> **Note**
>
> Hay más en GitHub. Busque el ejemplo completo y aprenda a configurar y ejecutar en el Repositorio de ejemplos de código de AWS.

Establecimiento de la configuración de bloqueo de objetos de 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()
+ "'");
       }
   }
```

Establecimiento del período de retención predeterminado de 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
+".");
    }
```

- Para obtener información sobre la API, consulte PutObjectLockConfiguration en la Referencia de la API de AWS SDK for Java 2.x.

JavaScript

SDK para JavaScript (v3)

> **Note**
>
> Hay más información en GitHub. Busque el ejemplo completo y aprenda a configurar y ejecutar en el Repositorio de ejemplos de código de AWS.

Establecimiento de la configuración de bloqueo de objetos de 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");
}
```

Establecimiento del período de retención predeterminado de 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");
}
```

- Para obtener información sobre la API, consulte PutObjectLockConfiguration en la Referencia de la API de AWS SDK for JavaScript.

Python

SDK para Python (Boto3)

> **Note**
>
> Hay más en GitHub. Busque el ejemplo completo y aprenda a configurar y ejecutar en el Repositorio de ejemplos de código de AWS.

Configuración de la activación del bloqueo de objetos.

```
        s3_client.put_object_lock_configuration(
            Bucket=bucket,
            ObjectLockConfiguration={"ObjectLockEnabled": "Disabled", "Rule":
 {}},
        )
```

- Para obtener más información sobre la API, consulte PutObjectLockConfiguration en la Referencia de la API de AWS SDK para Python (Boto3).

Para obtener una lista completa de las guías para desarrolladores del AWS SDK y ejemplos de código, consulte Uso de este servicio con un SDK de AWS. En este tema también se incluye información sobre cómo comenzar a utilizar el SDK y detalles sobre sus versiones anteriores.

Uso de **`PutObjectRetention`** con un AWS SDK o la CLI

Los siguientes ejemplos de código muestran cómo utilizar `PutObjectRetention`.

Los ejemplos de acciones son extractos de código de programas más grandes y deben ejecutarse en contexto. Puede ver esta acción en contexto en el siguiente ejemplo de código:

- Bloqueo de objetos de Amazon S3

.NET

AWS SDK for .NET

> **Note**
>
> Hay más información en GitHub. Busque el ejemplo completo y aprenda a configurar y ejecutar en el Repositorio de ejemplos de código de 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;
        }
    }
```

- Para obtener información sobre la API, consulte PutObjectRetention en la Referencia de la API de AWS SDK for .NET.

CLI

AWS CLI

Para establecer la configuración de retención de un objeto

En el siguiente ejemplo de `put-object-retention`, se establece una configuración de retención del objeto especificado hasta el 1 de enero de 2025.

```
aws s3api put-object-retention \
    --bucket my-bucket-with-object-lock \
    --key doc1.rtf \
    --retention '{ "Mode": "GOVERNANCE", "RetainUntilDate":
 "2025-01-01T00:00:00" }'
```

Este comando no genera ninguna salida.

- Para obtener información sobre la API, consulte PutObjectRetention en la Referencia de comandos de la AWS CLI.

Go

SDK para Go V2

> **Note**
>
> Hay más en GitHub. Busque el ejemplo completo y aprenda a configurar y ejecutar en el Repositorio de ejemplos de código de 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
}
```

- Para obtener información sobre la API, consulte PutObjectRetention en la Referencia de la API de AWS SDK for Go.

Java

SDK para Java 2.x

> **Note**
>
> Hay más en GitHub. Busque el ejemplo completo y aprenda a configurar y ejecutar en el Repositorio de ejemplos de código de 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 +".");
    }
```

- Para obtener información sobre la API, consulte PutObjectRetention en la Referencia de la API de AWS SDK for Java 2.x.

JavaScript

SDK para JavaScript (v3)

> **Note**
>
> Hay más información en GitHub. Busque el ejemplo completo y aprenda a configurar y ejecutar en el Repositorio de ejemplos de código de 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");
}
```

- Para obtener información sobre la API, consulte PutObjectRetention en la Referencia de la API de AWS SDK for JavaScript.

PowerShell

Herramientas para PowerShell

Ejemplo 1: el comando habilita el modo de retención de gobierno hasta la fecha “31 de diciembre de 2019 a las 00:00:00” para el objeto “testfile.txt” del bucket de S3 indicado.

```
Write-S3ObjectRetention -BucketName 's3buckettesting' -Key 'testfile.txt' -
Retention_Mode GOVERNANCE -Retention_RetainUntilDate "2019-12-31T00:00:00"
```

- Para obtener información sobre la API, consulte PutObjectRetention en la Referencia de Cmdlet de AWS Tools for PowerShell.

Python

SDK para Python (Boto3)

> **Note**
>
> Hay más en GitHub. Busque el ejemplo completo y aprenda a configurar y ejecutar en el Repositorio de ejemplos de código de AWS.

Ponga un retención de objeto.

```
            s3_client.put_object_retention(
                Bucket=bucket,
                Key=key,
                VersionId=version_id,
                Retention={"Mode": "GOVERNANCE", "RetainUntilDate":
far_future_date},
                BypassGovernanceRetention=True,
            )
```

- Para obtener información sobre la API, consulte PutObjectRetention en la Referencia de la API de AWS SDK para Python (Boto3).

Para obtener una lista completa de las guías para desarrolladores del AWS SDK y ejemplos de código, consulte Uso de este servicio con un SDK de AWS. En este tema también se incluye información sobre cómo comenzar a utilizar el SDK y detalles sobre sus versiones anteriores.

Uso de **`RestoreObject`** con un AWS SDK o la CLI

Los siguientes ejemplos de código muestran cómo utilizar `RestoreObject`.

.NET

AWS SDK for .NET

> **Note**
>
> Hay más en GitHub. Busque el ejemplo completo y aprenda a configurar y ejecutar en el Repositorio de ejemplos de código de 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}");
        }
    }
```

- Para obtener información sobre la API, consulte RestoreObject en la Referencia de la API de AWS SDK for .NET.

CLI

AWS CLI

Creación de una solicitud de restauración para un objeto

En el siguiente ejemplo de `restore-object` se restaura el objeto de Amazon S3 Glacier especificado para el bucket `my-glacier-bucket` durante 10 días.

```
aws s3api restore-object \
    --bucket my-glacier-bucket \
    --key doc1.rtf \
    --restore-request Days=10
```

Este comando no genera ninguna salida.

- Para obtener información sobre la API, consulte RestoreObject en la Referencia de comandos de la AWS CLI.

Java

SDK para Java 2.x

> **Note**
>
> Hay más en GitHub. Busque el ejemplo completo y aprenda a configurar y ejecutar en el Repositorio de ejemplos de código de 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);
        }
    }
```

```
}
```

- Para obtener información sobre la API, consulte RestoreObject en la Referencia de la API de AWS SDK for Java 2.x.

Para obtener una lista completa de las guías para desarrolladores del AWS SDK y ejemplos de código, consulte Uso de este servicio con un SDK de AWS. En este tema también se incluye información sobre cómo comenzar a utilizar el SDK y detalles sobre sus versiones anteriores.

Uso de **`SelectObjectContent`** con un AWS SDK o la CLI

Los siguientes ejemplos de código muestran cómo utilizar `SelectObjectContent`.

CLI

AWS CLI

Para filtrar el contenido de un objeto de Amazon S3 en función de una instrucción SQL

En el siguiente ejemplo de `select-object-content`, se filtra el objeto `my-data-file.csv` con la instrucción SQL especificada y se envía el resultado a un archivo.

```
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"
```

Este comando no genera ninguna salida.

- Para obtener información sobre la API, consulte SelectObjectContent en la Referencia de comandos de la AWS CLI.

Java

SDK para Java 2.x

> **Note**
>
> Hay más en GitHub. Busque el ejemplo completo y aprenda a configurar y ejecutar en el Repositorio de ejemplos de código de AWS.

En el siguiente ejemplo, se muestra una consulta utilizando un objeto JSON. En el ejemplo completo, también se muestra el uso de un objeto 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;
        }
    }
```

- Para obtener información sobre la API, consulte SelectObjectContent en la Referencia de la API de AWS SDK for Java 2.x.

Para obtener una lista completa de las guías para desarrolladores del AWS SDK y ejemplos de código, consulte Uso de este servicio con un SDK de AWS. En este tema también se incluye información sobre cómo comenzar a utilizar el SDK y detalles sobre sus versiones anteriores.

Uso de **`UploadPart`** con un AWS SDK o la CLI

Los siguientes ejemplos de código muestran cómo utilizar `UploadPart`.

Los ejemplos de acciones son extractos de código de programas más grandes y deben ejecutarse en contexto. Puede ver esta acción en su contexto en los siguientes ejemplos de código:

- Ejecución de una carga multiparte
- Usar sumas de comprobación

- Trabajo con la integridad de los objetos de Amazon S3

C++

SDK para C++

> **Note**
>
> Hay más en GitHub. Busque el ejemplo completo y aprenda a configurar y ejecutar en el Repositorio de ejemplos de código de AWS.

```
//! Upload a part to an S3 bucket.
/*!
    \param bucket: The name of the S3 bucket where the object will be uploaded.
    \param key: The unique identifier (key) for the object within the S3 bucket.
    \param uploadID: An upload ID string.
    \param partNumber:
    \param checksumAlgorithm: Checksum algorithm, ignored when NOT_SET.
    \param calculatedHash: A data integrity hash to set, depending on the
 checksum algorithm,
                           ignored when it is an empty string.
    \param body: An shared_ptr IOStream of the data to be uploaded.
    \param client: The S3 client instance used to perform the upload operation.
    \return UploadPartOutcome: The outcome.
*/

Aws::S3::Model::UploadPartOutcome AwsDoc::S3::uploadPart(const Aws::String
 &bucket,
                                                         const Aws::String &key,
                                                         const Aws::String
 &uploadID,
                                                         int partNumber,

 Aws::S3::Model::ChecksumAlgorithm checksumAlgorithm,
                                                         const Aws::String
 &calculatedHash,
                                                         const
 std::shared_ptr<Aws::IOStream> &body,
                                                         const Aws::S3::S3Client
 &client) {
    Aws::S3::Model::UploadPartRequest request;
```

```
    request.SetBucket(bucket);
    request.SetKey(key);
    request.SetUploadId(uploadID);
    request.SetPartNumber(partNumber);
    if (checksumAlgorithm != Aws::S3::Model::ChecksumAlgorithm::NOT_SET) {
        request.SetChecksumAlgorithm(checksumAlgorithm);
    }
    request.SetBody(body);

    if (!calculatedHash.empty()) {
        switch (checksumAlgorithm) {
            case Aws::S3::Model::ChecksumAlgorithm::NOT_SET:
                request.SetContentMD5(calculatedHash);
                break;
            case Aws::S3::Model::ChecksumAlgorithm::CRC32:
                request.SetChecksumCRC32(calculatedHash);
                break;
            case Aws::S3::Model::ChecksumAlgorithm::CRC32C:
                request.SetChecksumCRC32C(calculatedHash);
                break;
            case Aws::S3::Model::ChecksumAlgorithm::SHA1:
                request.SetChecksumSHA1(calculatedHash);
                break;
            case Aws::S3::Model::ChecksumAlgorithm::SHA256:
                request.SetChecksumSHA256(calculatedHash);
                break;
        }
    }

    return client.UploadPart(request);
}
```

- Para obtener información sobre la API, consulte UploadPart en la Referencia de la API de AWS SDK for C++.

CLI

AWS CLI

El siguiente comando carga la primera parte de una carga multiparte iniciada con el comando `create-multipart-upload`:

```
aws s3api upload-part --bucket my-bucket --key 'multipart/01' --part-number 1 --
body part01 --upload-id
  "dfRtDYU0WWCCcH43C3WFbkRONycyCpTJJvxu2i5GYkZljF.Yxwh6XG7WfS2vC4to6HiV6Yjlx.cph0gtNBtJ8P
```

La opción body toma el nombre o la ruta de un archivo local para la carga (no utilice el prefijo file://). El tamaño mínimo de parte es de 5 MB. El ID de carga lo devuelve `create-multipart-upload` y también se puede recuperar con `list-multipart-uploads`. El bucket y la clave se especifican al crear la carga multiparte.

Salida:

```
{
    "ETag": "\"e868e0f4719e394144ef36531ee6824c\""
}
```

Guarde el valor de ETag de cada parte para más adelante. Son necesarios para completar la carga multiparte.

- Para obtener información sobre la API, consulte UploadPart en la Referencia de comandos de la AWS CLI.

Rust

SDK para Rust

> **Note**
>
> Hay más información en GitHub. Busque el ejemplo completo y aprenda a configurar y ejecutar en el Repositorio de ejemplos de código de 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();
```

- Para obtener detalles sobre la API, consulte UploadPart en la Referencia de la API del SDK de AWS para Rust.

Para obtener una lista completa de las guías para desarrolladores del AWS SDK y ejemplos de código, consulte Uso de este servicio con un SDK de AWS. En este tema también se incluye información sobre cómo comenzar a utilizar el SDK y detalles sobre sus versiones anteriores.

Escenarios de Amazon S3 con SDK de AWS

Los siguientes ejemplos de código muestran cómo implementar escenarios comunes en Amazon S3 con el SDK AWS. Estos escenarios muestran cómo llevar a cabo tareas específicas llamando a varias funciones dentro de Amazon S3. En cada escenario se incluye un enlace a GitHub, con instrucciones de configuración y ejecución del código.

Ejemplos

- Cree una URL prefirmada para Amazon S3 mediante un SDK de AWS
- Una página web que indica los objetos de Amazon S3 que usan un SDK de AWS
- Eliminación de las cargas multiparte incompletas a Amazon S3 mediante un AWS SDK
- Descargar todos los objetos de un bucket de Amazon Simple Storage Service (Amazon S3) en un directorio local
- Obtención de un objeto de Amazon S3 desde un punto de acceso de varias regiones con un SDK de AWS
- Obtenga un objeto de un bucket de Amazon S3 con un SDK de AWS al especificar un encabezado If-Modified-Since

- Introducción a los buckets y objetos de Amazon S3 con un SDK de AWS
- Introducción al cifrado de objetos de Amazon S3 con un SDK de AWS
- Introducción a etiquetas de objetos de Amazon S3 con un SDK de AWS
- Obtención de la configuración de retención legal de un objeto de Amazon S3 mediante un SDK de AWS
- Trabajo con las características de bloqueo de objetos de Amazon S3 mediante un SDK de AWS
- Administre listas de control de acceso (ACL) para buckets de Amazon S3 con un SDK de AWS
- Administre objetos de Amazon S3 con control de versiones en lotes con una función de Lambda mediante un SDK de AWS
- Analizar los URI de Amazon S3 mediante un SDK de AWS
- Ejecución de una copia multiparte de un objeto de Amazon S3 con un SDK de AWS
- Ejecución de una carga multiparte de un objeto de Amazon S3 con un AWS SDK
- Reciba y procese las notificaciones de eventos de Amazon S3 mediante un AWS SDK.
- Envío de notificaciones de eventos de S3 a Amazon EventBridge mediante un AWS SDK
- Realización de un seguimiento de la carga o descarga de un objeto de Amazon S3 mediante un AWS SDK
- Ejemplos de enfoques para pruebas unitarias y de integración con un SDK de AWS
- Cargar de forma recursiva un directorio local en un bucket de Amazon Simple Storage Service (Amazon S3)
- Cargar o descargar archivos de gran tamaño desde y hacia Amazon S3 con un SDK de AWS
- Carga de un flujo de tamaño desconocido en un objeto de Amazon S3 mediante un SDK de AWS
- Uso de sumas de comprobación para trabajar con un objeto de Amazon S3 con un SDK de AWS
- Trabajo con las características de integridad de objetos de Amazon S3 utilizando un AWS SDK
- Trabajo con objetos con control de versiones de Amazon S3 con un SDK de AWS

Cree una URL prefirmada para Amazon S3 mediante un SDK de AWS

En los siguientes ejemplos de código, se muestra cómo crear una URL prefirmada para Amazon S3 y cargar un objeto.

.NET

AWS SDK for .NET

> **Note**
>
> Hay más en GitHub. Busque el ejemplo completo y aprenda a configurar y ejecutar en el Repositorio de ejemplos de código de AWS.

Genere una URL prefirmada que pueda realizar una acción de Amazon S3 durante un tiempo limitado.

```
    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;
        }
    }
```

Genere una URL prefirmada y realice una carga con esa 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 para C++

> **Note**
>
> Hay más en GitHub. Busque el ejemplo completo y aprenda a configurar y ejecutar en el Repositorio de ejemplos de código de AWS.

Genere una URL prefirmada para descargar un objeto.

```
//! 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);
}
```

Descargue mediante 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;
```

```
}
```

Genere una URL prefirmada para cargar un objeto.

```
//! 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);
}
```

Cargue mediante 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

SDK para Go V2

> **Note**
>
> Hay más en GitHub. Busque el ejemplo completo y aprenda a configurar y ejecutar en el Repositorio de ejemplos de código de AWS.

Crear funciones que encapsulen acciones prefirma de 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
}
```

Ejecute un ejemplo interactivo que genere y utilice URL prefirmadas para cargar, descargar y eliminar un objeto de 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))
}
```

Defina un contenedor de solicitudes HTTP utilizado en el ejemplo para realizar solicitudes 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 para Java 2.x

> **Note**
>
> Hay más en GitHub. Busque el ejemplo completo y aprenda a configurar y ejecutar en el Repositorio de ejemplos de código de AWS.

Genere una URL prefirmada para un objeto y, a continuación, descárguela (solicitud GET).

Importaciones.

```
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;
```

Genere la 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();
        }
    }
```

Descargue el objeto mediante uno de los tres enfoques siguientes.

Utilice la clase JDK `HttpURLConnection` (desde la versión 1.1) para realizar la descarga.

```
    /* 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();
    }
```

Utilice la clase JDK `HttpClient` (desde la versión 11) para realizar la descarga.

```
    /* 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();
   }
```

Utilice el SDK de AWS para la clase `SdkHttpClient` de Java para realizar la descarga.

```
   /* 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();
    }
```

Genere una URL prefirmada para una carga y, a continuación, cargue un archivo (solicitud PUT).

Importaciones.

```
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;
```

Genere la 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();
        }
    }
```

Cargue un objeto de archivo mediante uno de los tres enfoques siguientes.

Utilice la clase JDK `HttpURLConnection` (desde la versión 1.1) para realizar la carga.

```
    /* 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);
        }
    }
```

Utilice la clase JDK `HttpClient` (desde la versión 11) para realizar la carga.

```
    /* 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);
        }
    }
```

Utilice la clase `SdkHttpClient` de AWS para Java V2 para realizar la carga.

```
   /* 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 para JavaScript (v3)

> **Note**
>
> Hay más información en GitHub. Busque el ejemplo completo y aprenda a configurar y ejecutar en el Repositorio de ejemplos de código de AWS.

Cree una URL prefirmada para cargar un objeto en un bucket.

```
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);
  }
};
```

Cree una URL prefirmada para descargar un objeto de un bucket.

```
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);
  }
};
```

- Para obtener información, consulte la Guía para desarrolladores de AWS SDK for JavaScript.

Kotlin

SDK para Kotlin

> **Note**
>
> Hay más información en GitHub. Busque el ejemplo completo y aprenda a configurar y ejecutar en el Repositorio de ejemplos de código de AWS.

Cree una solicitud prefirmada de `GetObject` y utilice la URL para descargar un objeto.

```
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
}
```

Cree una solicitud prefirmada `GetObject` con opciones avanzadas.

```
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
}
```

Cree una solicitud prefirmada de `PutObject` y úsela para subir un objeto.

```
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)
}
```

- Para obtener información, consulte Guía para desarrolladores del SDK de AWS SDK para Kotlin.

PHP

SDK para PHP

> **Note**
>
> Hay más en GitHub. Busque el ejemplo completo y aprenda a configurar y ejecutar en el Repositorio de ejemplos de código de 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 para Python (Boto3)

> **Note**
>
> Hay más en GitHub. Busque el ejemplo completo y aprenda a configurar y ejecutar en el Repositorio de ejemplos de código de AWS.

Genere una URL prefirmada que pueda realizar una acción de S3 durante un tiempo limitado. Utilice el paquete Requests para realizar una solicitud con la 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()
```

Genere una solicitud POST prefirmada para cargar un archivo.

```
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

SDK para Ruby

> **Note**
>
> Hay más en GitHub. Busque el ejemplo completo y aprenda a configurar y ejecutar en el Repositorio de ejemplos de código de 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 para Rust

> **Note**
>
> Hay más información en GitHub. Busque el ejemplo completo y aprenda a configurar y ejecutar en el Repositorio de ejemplos de código de AWS.

Cree solicitudes de prefirma para objetos GET y PUT S3.

```
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(())
}
```

Para obtener una lista completa de las guías para desarrolladores del AWS SDK y ejemplos de código, consulte Uso de este servicio con un SDK de AWS. En este tema también se incluye información sobre cómo comenzar a utilizar el SDK y detalles sobre sus versiones anteriores.

Una página web que indica los objetos de Amazon S3 que usan un SDK de AWS

En los siguientes ejemplos de código se muestra cómo obtener una lista de los objetos de Amazon S3 en una página web.

JavaScript

SDK para JavaScript (v3)

> **Note**
>
> Hay más información en GitHub. Busque el ejemplo completo y aprenda a configurar y ejecutar en el Repositorio de ejemplos de código de AWS.

El siguiente código es el componente de React relevante que realiza llamadas al SDK de AWS. Hay una versión ejecutable de la aplicación que contiene este componente en el enlace anterior de 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;
```

- Para obtener información sobre la API, consulte ListObjects en la Referencia de la API de AWS SDK for JavaScript.

Para obtener una lista completa de las guías para desarrolladores del AWS SDK y ejemplos de código, consulte Uso de este servicio con un SDK de AWS. En este tema también se incluye información sobre cómo comenzar a utilizar el SDK y detalles sobre sus versiones anteriores.

Eliminación de las cargas multiparte incompletas a Amazon S3 mediante un AWS SDK

En el siguiente ejemplo de código, se muestra cómo eliminar o detener cargas multiparte incompletas de Amazon S3.

Java

SDK para Java 2.x

> **Note**
>
> Hay más en GitHub. Busque el ejemplo completo y aprenda a configurar y ejecutar en el Repositorio de ejemplos de código de AWS.

Para detener las cargas multiparte que estén en curso o incompletas por cualquier motivo, puede obtener una lista de las cargas y, a continuación, eliminarlas, tal y como se muestra en el siguiente ejemplo.

```
   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);
            }
        }
    }
```

Para eliminar las cargas multiparte incompletas que se hayan iniciado antes o después de una fecha, puede eliminar de forma selectiva las cargas multiparte en función de un momento dado, como se muestra en el siguiente ejemplo.

```
    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 tiene acceso al ID de carga después de iniciar una carga multiparte, puede eliminar la carga en curso utilizando ese ID.

```
    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);
        }
    }
```

Para eliminar de forma sistemática las cargas multiparte incompletas que tengan más de un número determinado de días, establezca una configuración de ciclo de vida de bucket para el bucket. En el siguiente ejemplo, se muestra cómo crear una regla para eliminar cargas incompletas que tienen más de 7 días.

```
    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());
        }
    }
```

- Para obtener información sobre la API, consulte los siguientes temas en la referencia de la API de AWS SDK for Java 2.x.
  - AbortMultipartUpload
  - ListMultipartUploads
  - PutBucketLifecycleConfiguration

Para obtener una lista completa de las guías para desarrolladores del AWS SDK y ejemplos de código, consulte Uso de este servicio con un SDK de AWS. En este tema también se incluye información sobre cómo comenzar a utilizar el SDK y detalles sobre sus versiones anteriores.

Descargar todos los objetos de un bucket de Amazon Simple Storage Service (Amazon S3) en un directorio local

El siguiente ejemplo de código muestra cómo descargar todos los objetos de un bucket de Amazon Simple Storage Service (Amazon S3) en un directorio local.

Java

SDK para Java 2.x

> **Note**
>
> Hay más en GitHub. Busque el ejemplo completo y aprenda a configurar y ejecutar en el Repositorio de ejemplos de código de AWS.

Utilice un S3TransferManager para descargar todos los objetos de S3 en el mismo bucket de S3. Vea el archivo completo y pruébelo.

```
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();
    }
```

- Para obtener información acerca de la API, consulte DownloadDirectory en la ReferenciaAWS SDK for Java 2.x de la API de.

Para obtener una lista completa de las guías para desarrolladores del AWS SDK y ejemplos de código, consulte Uso de este servicio con un SDK de AWS. En este tema también se incluye información sobre cómo comenzar a utilizar el SDK y detalles sobre sus versiones anteriores.

Obtención de un objeto de Amazon S3 desde un punto de acceso de varias regiones con un SDK de AWS

En el siguiente ejemplo de código se muestra cómo obtener un objeto desde un punto de acceso de varias regiones.

Kotlin

SDK para Kotlin

> **Note**
>
> Hay más información en GitHub. Busque el ejemplo completo y aprenda a configurar y ejecutar en el Repositorio de ejemplos de código de AWS.

Configure el cliente S3 para que utilice el algoritmo de firma asimétrica 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
        }
```

Utilice el ARN del punto de acceso de varias regiones en lugar del nombre de un bucket para obtener el objeto.

```
    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
    }
```

- Para obtener información, consulte Guía para desarrolladores del SDK de AWS SDK para Kotlin.
- Para obtener información sobre la API, consulte GetObject en la Referencia de la API de SDK de AWS para Kotlin.

Para obtener una lista completa de las guías para desarrolladores del AWS SDK y ejemplos de código, consulte Uso de este servicio con un SDK de AWS. En este tema también se incluye información sobre cómo comenzar a utilizar el SDK y detalles sobre sus versiones anteriores.

Obtenga un objeto de un bucket de Amazon S3 con un SDK de AWS al especificar un encabezado If-Modified-Since

En el siguiente ejemplo de código se muestra cómo leer datos de un objeto en un bucket de S3, pero solo si ese bucket no se ha modificado desde la última recuperación.

Rust

SDK para Rust

> **Note**
>
> Hay más información en GitHub. Busque el ejemplo completo y aprenda a configurar y ejecutar en el Repositorio de ejemplos de código de 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(())
}
```

- Para obtener información sobre la API, consulte GetObject en la Referencia de la API de AWS SDK para Rust.

Para obtener una lista completa de las guías para desarrolladores del AWS SDK y ejemplos de código, consulte Uso de este servicio con un SDK de AWS. En este tema también se incluye información sobre cómo comenzar a utilizar el SDK y detalles sobre sus versiones anteriores.

Introducción a los buckets y objetos de Amazon S3 con un SDK de AWS

En el siguiente ejemplo de código, se muestra cómo:

- Crear un bucket y cargar un archivo en el bucket.
- Descargar un objeto desde un bucket.
- Copiar un objeto en una subcarpeta de un bucket.
- Obtención de una lista de los objetos de un bucket.
- Eliminar el bucket y todos los objetos que incluye.

.NET

AWS SDK for .NET

> **Note**
>
> Hay más en GitHub. Busque el ejemplo completo y aprenda a configurar y ejecutar en el Repositorio de ejemplos de código de 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);
        }
    }
```

- Para obtener información sobre la API, consulte los siguientes temas en la referencia de la API de AWS SDK for .NET.
  - CopyObject
  - CreateBucket
  - DeleteBucket
  - DeleteObjects
  - GetObject
  - ListObjectsV2
  - PutObject

Bash

AWS CLI con script Bash

> **Note**
>
> Hay más información en GitHub. Busque el ejemplo completo y aprenda a configurar y ejecutar en el Repositorio de ejemplos de código de 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
}
```

Las funciones de Amazon S3 utilizadas en este escenario.

```
###############################################################################
# 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
}
```

- Para obtener información sobre la API, consulte los siguientes temas en la Referencia de comandos de AWS CLI.
  - CopyObject
  - CreateBucket
  - DeleteBucket
  - DeleteObjects
  - GetObject
  - ListObjectsV2
  - PutObject

C++

SDK para C++

> **Note**
>
> Hay más en GitHub. Busque el ejemplo completo y aprenda a configurar y ejecutar en el Repositorio de ejemplos de código de 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();
}
```

- Para obtener información sobre la API, consulte los siguientes temas en la referencia de la API de AWS SDK for C++.
  - CopyObject

- CreateBucket
- DeleteBucket
- DeleteObjects
- GetObject
- ListObjectsV2
- PutObject

Go

SDK para Go V2

> **Note**
>
> Hay más en GitHub. Busque el ejemplo completo y aprenda a configurar y ejecutar en el Repositorio de ejemplos de código de AWS.

Defina una estructura que envuelva las acciones de bucket y objeto utilizadas por el escenario.

```
// 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
}
```

Ejecute un escenario interactivo que le muestre cómo trabajar con buckets y objetos de 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))
}
```

- Para obtener información sobre la API, consulte los siguientes temas en la referencia de la API de AWS SDK for Go.
  - CopyObject
  - CreateBucket
  - DeleteBucket
  - DeleteObjects
  - GetObject
  - ListObjectsV2
  - PutObject

Java

SDK para Java 2.x

> **Note**
>
> Hay más en GitHub. Busque el ejemplo completo y aprenda a configurar y ejecutar en el Repositorio de ejemplos de código de 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 "";
    }
}
```

- Para obtener información sobre la API, consulte los siguientes temas en la referencia de la API de AWS SDK for Java 2.x.
  - CopyObject
  - CreateBucket
  - DeleteBucket
  - DeleteObjects
  - GetObject
  - ListObjectsV2
  - PutObject

JavaScript

SDK para JavaScript (v3)

> **Note**
>
> Hay más información en GitHub. Busque el ejemplo completo y aprenda a configurar y ejecutar en el Repositorio de ejemplos de código de AWS.

Primero, importe todos los módulos necesarios.

```
// 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";
```

Las importaciones anteriores hacen referencia a algunas utilidades auxiliares. Estas utilidades son locales del repositorio de GitHub vinculado al principio de esta sección. Consulte las siguientes implementaciones de esas utilidades a modo de referencia.

```
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`;
};
```

Los objetos de S3 se almacenan en buckets (cubos). Definamos una función para crear un nuevo bucket.

```
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;
};
```

Los buckets contienen «objetos». Esta función carga el contenido de un directorio al bucket en forma de objetos.

```
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.`);
  }
};
```

Después de cargar los objetos, confirme que se hayan subido correctamente. Puede usar ListObjects para la comprobación. Utilizará la propiedad «Clave», pero también hay otras propiedades útiles en la respuesta.

```
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");
};
```

A veces, es posible que quiera copiar un objeto de un bucket en otro bucket. Para ello, utilice el comando CopyObject.

```
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();
  }
};
```

No existe ningún método de SDK para obtener varios objetos de un bucket. En su lugar, creará una lista de objetos para descargarlos e iterarlos.

```
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");
};
```

Ha llegado el momento de limpiar los recursos. Un bucket debe estar vacío para poder eliminarlo. Estas dos funciones vacían y eliminan el bucket.

```
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 función «principal» reúne todo. Si ejecuta este archivo directamente, se llamará a la función principal.

```
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);
  }
};
```

- Para obtener información sobre la API, consulte los siguientes temas en la referencia de la API de AWS SDK for JavaScript.
  - CopyObject
  - CreateBucket

- DeleteBucket
- DeleteObjects
- GetObject
- ListObjectsV2
- PutObject

Kotlin

SDK para Kotlin

> **Note**
>
> Hay más información en GitHub. Busque el ejemplo completo y aprenda a configurar y ejecutar en el Repositorio de ejemplos de código de 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!")
    }
}
```

- Para obtener información sobre la API, consulte los siguientes temas en la Referencia de la API del SDKde AWS para Kotlin.
  - CopyObject
  - CreateBucket
  - DeleteBucket
  - DeleteObjects
  - GetObject
  - ListObjectsV2
  - PutObject

PHP

SDK para PHP

> **Note**
>
> Hay más en GitHub. Busque el ejemplo completo y aprenda a configurar y ejecutar en el Repositorio de ejemplos de código de 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";
```

- Para obtener información sobre la API, consulte los siguientes temas en la referencia de la API de AWS SDK for PHP.
  - CopyObject
  - CreateBucket
  - DeleteBucket
  - DeleteObjects
  - GetObject
  - ListObjectsV2
  - PutObject

Python

SDK para Python (Boto3)

> **Note**
>
> Hay más en GitHub. Busque el ejemplo completo y aprenda a configurar y ejecutar en el Repositorio de ejemplos de código de 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"))
```

- Para obtener información sobre la API, consulte los siguientes temas en la Referencia de la API de AWS SDK para Python (Boto3).
  - CopyObject
  - CreateBucket
  - DeleteBucket
  - DeleteObjects
  - GetObject
  - ListObjectsV2
  - PutObject

Ruby

SDK para Ruby

> **Note**
>
> Hay más en GitHub. Busque el ejemplo completo y aprenda a configurar y ejecutar en el Repositorio de ejemplos de código de 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__
```

- Para obtener información sobre la API, consulte los siguientes temas en la referencia de la API de AWS SDK for Ruby.
  - CopyObject
  - CreateBucket
  - DeleteBucket
  - DeleteObjects
  - GetObject
  - ListObjectsV2
  - PutObject

Rust

SDK para Rust

> **Note**
>
> Hay más información en GitHub. Busque el ejemplo completo y aprenda a configurar y ejecutar en el Repositorio de ejemplos de código de AWS.

Código para la caja binaria que ejecuta el escenario.

```
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(())
}
```

Caja de biblioteca con acciones comunes que llama el binario.

```
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
```

```
}
```

- Para obtener información sobre la API, consulte los siguientes temas en la Referencia de la API de AWS SDK para Rust.
  - CopyObject
  - CreateBucket
  - DeleteBucket
  - DeleteObjects
  - GetObject
  - ListObjectsV2
  - PutObject

SAP ABAP

SDK de SAP ABAP

> **Note**
>
> Hay más en GitHub. Busque el ejemplo completo y aprenda a configurar y ejecutar en el Repositorio de ejemplos de código de 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.
```

- Para detalles acerca de la API, consulte los siguientes temas en la Referencia de la API del SDK de AWS para SAP ABAP.
  - CopyObject
  - CreateBucket
  - DeleteBucket
  - DeleteObjects
  - GetObject
  - ListObjectsV2
  - PutObject

Swift

SDK para Swift

> **Note**
>
> Esto es documentación preliminar para un SDK en versión preliminar. Está sujeta a cambios.

> **Note**
>
> Hay más en GitHub. Busque el ejemplo completo y aprenda a configurar y ejecutar en el Repositorio de ejemplos de código de AWS.

Una clase de Swift que gestiona las llamadas al SDK para Swift.

```
import Foundation
import AWSS3
import ClientRuntime
import AWSClientRuntime
import Smithy
```

```
/// 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.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.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 programa de línea de comandos de Swift para administrar las llamadas al 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)
```

```
        }
    }
}
```

- Para obtener información sobre la API, consulte los siguientes temas en la Referencia de la API de AWS SDK para Swift.
  - CopyObject
  - CreateBucket
  - DeleteBucket
  - DeleteObjects
  - GetObject
  - ListObjectsV2
  - PutObject

Para obtener una lista completa de las guías para desarrolladores del AWS SDK y ejemplos de código, consulte Uso de este servicio con un SDK de AWS. En este tema también se incluye información sobre cómo comenzar a utilizar el SDK y detalles sobre sus versiones anteriores.

Introducción al cifrado de objetos de Amazon S3 con un SDK de AWS

El siguiente ejemplo de código muestra cómo empezar a cifrar objetos de Amazon S3.

.NET

AWS SDK for .NET

> **Note**
>
> Hay más en GitHub. Busque el ejemplo completo y aprenda a configurar y ejecutar en el Repositorio de ejemplos de código de 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);
        }
    }
```

- Para obtener información sobre la API, consulte los siguientes temas en la referencia de la API de AWS SDK for .NET.
  - CopyObject
  - GetObject
  - GetObjectMetadata

Para obtener una lista completa de las guías para desarrolladores del AWS SDK y ejemplos de código, consulte Uso de este servicio con un SDK de AWS. En este tema también se incluye información sobre cómo comenzar a utilizar el SDK y detalles sobre sus versiones anteriores.

Introducción a etiquetas de objetos de Amazon S3 con un SDK de AWS

El ejemplo de código siguiente muestra cómo empezar a usar etiquetas para objetos de Amazon S3.

.NET

AWS SDK for .NET

> **Note**
>
> Hay más en GitHub. Busque el ejemplo completo y aprenda a configurar y ejecutar en el Repositorio de ejemplos de código de 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}'");
            }
        }
    }
```

- Para obtener información sobre la API, consulte GetObjectTagging en la Referencia de la API de AWS SDK for .NET.

Para obtener una lista completa de las guías para desarrolladores del AWS SDK y ejemplos de código, consulte Uso de este servicio con un SDK de AWS. En este tema también se incluye información sobre cómo comenzar a utilizar el SDK y detalles sobre sus versiones anteriores.

Obtención de la configuración de retención legal de un objeto de Amazon S3 mediante un SDK de AWS

Los siguientes ejemplos de código muestran cómo obtener la configuración de retención legal de un bucket de S3.

.NET

AWS SDK for .NET

> **Note**
>
> Hay más en GitHub. Busque el ejemplo completo y aprenda a configurar y ejecutar en el Repositorio de ejemplos de código de 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();
        }
    }
```

- Para obtener información sobre la API, consulte GetObjectLegalHold en la Referencia de la API de AWS SDK for .NET.

Java

SDK para Java 2.x

> **Note**
>
> Hay más en GitHub. Busque el ejemplo completo y aprenda a configurar y ejecutar en el Repositorio de ejemplos de código de 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;
    }
```

- Para obtener información sobre la API, consulte GetObjectLegalHold en la Referencia de la API de AWS SDK for Java 2.x.

JavaScript

SDK para JavaScript (v3)

> **Note**
>
> Hay más información en GitHub. Busque el ejemplo completo y aprenda a configurar y ejecutar en el Repositorio de ejemplos de código de 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");
}
```

- Para obtener información sobre la API, consulte GetObjectLegalHold en la Referencia de la API de AWS SDK for JavaScript.

Para obtener una lista completa de las guías para desarrolladores del AWS SDK y ejemplos de código, consulte Uso de este servicio con un SDK de AWS. En este tema también se incluye información sobre cómo comenzar a utilizar el SDK y detalles sobre sus versiones anteriores.

Trabajo con las características de bloqueo de objetos de Amazon S3 mediante un SDK de AWS

El siguiente ejemplo de código muestra cómo trabajar con características de bloqueo de objetos de S3.

.NET

AWS SDK for .NET

> **Note**
>
> Hay más en GitHub. Busque el ejemplo completo y aprenda a configurar y ejecutar en el Repositorio de ejemplos de código de AWS.

Ejecute un escenario interactivo en el que se demuestren las características de bloqueo de objetos de 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;
    }
}
```

Una clase contenedora para funciones de 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;
        }

    }

}
```

- Para obtener información sobre la API, consulte los siguientes temas en la referencia de la API de AWS SDK for .NET.
  - GetObjectLegalHold
  - GetObjectLockConfiguration
  - GetObjectRetention
  - PutObjectLegalHold
  - PutObjectLockConfiguration
  - PutObjectRetention

Go

SDK para Go V2

> **Note**
>
> Hay más en GitHub. Busque el ejemplo completo y aprenda a configurar y ejecutar en el Repositorio de ejemplos de código de AWS.

Ejecute un escenario interactivo en el que se demuestren las características de bloqueo de objetos de 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))
}
```

Defina una estructura que envuelva las acciones de S3 utilizadas en este ejemplo.

```
// 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
}
```

Eliminación de recursos.

```
// 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)
  }
 }
}
```

- Para obtener información sobre la API, consulte los siguientes temas en la referencia de la API de AWS SDK for Go.

- GetObjectLegalHold
- GetObjectLockConfiguration
- GetObjectRetention
- PutObjectLegalHold
- PutObjectLockConfiguration
- PutObjectRetention

Java

SDK para Java 2.x

> **Note**
>
> Hay más en GitHub. Busque el ejemplo completo y aprenda a configurar y ejecutar en el Repositorio de ejemplos de código de AWS.

Ejecute un escenario interactivo en el que se demuestren las características de bloqueo de objetos de 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;
    }
}
```

Una clase contenedora para funciones de 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());
    }
}
```

- Para obtener información sobre la API, consulte los siguientes temas en la referencia de la API de AWS SDK for Java 2.x.
  - GetObjectLegalHold
  - GetObjectLockConfiguration
  - GetObjectRetention
  - PutObjectLegalHold
  - PutObjectLockConfiguration
  - PutObjectRetention

JavaScript

SDK para JavaScript (v3)

> **Note**
>
> Hay más información en GitHub. Busque el ejemplo completo y aprenda a configurar y ejecutar en el Repositorio de ejemplos de código de AWS.

index.js: punto de entrada para el flujo de trabajo. Esto organiza todos los pasos. Visite GitHub para ver los detalles de implementación de Scenario, ScenarioInput, ScenarioOutput y 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: enviar mensajes de bienvenida a la consola.

```
// 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: implementar ajustes de buckets, objetos y archivos.

```
// 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: ver y eliminar archivos en los buckets.

```
// 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: destruir todos los recursos creados.

```
// 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 };
```

- Para obtener información sobre la API, consulte los siguientes temas en la referencia de la API de AWS SDK for JavaScript.
  - GetObjectLegalHold
  - GetObjectLockConfiguration
  - GetObjectRetention
  - PutObjectLegalHold
  - PutObjectLockConfiguration
  - PutObjectRetention

Para obtener una lista completa de las guías para desarrolladores del AWS SDK y ejemplos de código, consulte Uso de este servicio con un SDK de AWS. En este tema también se incluye información sobre cómo comenzar a utilizar el SDK y detalles sobre sus versiones anteriores.

Administre listas de control de acceso (ACL) para buckets de Amazon S3 con un SDK de AWS

En los siguientes ejemplos de código, se muestra cómo administrar listas de control de acceso (ACL) para buckets de Amazon S3.

.NET

AWS SDK for .NET

> **Note**
>
> Hay más en GitHub. Busque el ejemplo completo y aprenda a configurar y ejecutar en el Repositorio de ejemplos de código de 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,
            });
        }

    }
```

- Para obtener información sobre la API, consulte los siguientes temas en la referencia de la API de AWS SDK for .NET.
  - GetBucketAcl

- GetObjectAcl
- PutBucketAcl
- PutObjectAcl

Para obtener una lista completa de las guías para desarrolladores del AWS SDK y ejemplos de código, consulte Uso de este servicio con un SDK de AWS. En este tema también se incluye información sobre cómo comenzar a utilizar el SDK y detalles sobre sus versiones anteriores.

Administre objetos de Amazon S3 con control de versiones en lotes con una función de Lambda mediante un SDK de AWS

En el siguiente ejemplo de código, se muestra cómo administrar objetos de S3 con control de versiones en lotes con una función de Lambda.

Python

SDK para Python (Boto3)

Muestra cómo manipular los objetos con control de versiones de Amazon Simple Storage Service (Amazon S3) en lotes mediante la creación de trabajos que llaman a funciones AWS Lambda para llevar a cabo el procesamiento. En este ejemplo se crea un bucket con control de versiones, se cargan las estrofas del poema You Are Old, Father William de Lewis Carroll y se utilizan trabajos por lotes de Amazon S3 para cambiar el poema de varias formas.

Aprenda cómo:

- Crear funciones Lambda que funcionen en objetos con control de versiones.
- Crear un manifiesto de objetos para actualizar.
- Crear trabajos por lotes que invoquen funciones Lambda para actualizar objetos.
- Eliminar funciones Lambda.
- Vaciar y eliminar un bucket con control de versiones.

Este ejemplo se puede ver mejor en GitHub. Para ver el código fuente completo y las instrucciones de configuración y ejecución, consulte el ejemplo completo en GitHub.

Servicios utilizados en este ejemplo

- Amazon S3

Para obtener una lista completa de las guías para desarrolladores del AWS SDK y ejemplos de código, consulte Uso de este servicio con un SDK de AWS. En este tema también se incluye información sobre cómo comenzar a utilizar el SDK y detalles sobre sus versiones anteriores.

Analizar los URI de Amazon S3 mediante un SDK de AWS

En el siguiente ejemplo se muestra cómo analizar los URI de Amazon S3 para extraer componentes importantes como el nombre del bucket y la clave de objeto.

Java

SDK para Java 2.x

> **Note**
>
> Hay más en GitHub. Busque el ejemplo completo y aprenda a configurar y ejecutar en el Repositorio de ejemplos de código de AWS.

Analice un URI de Amazon S3 mediante la clase 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);
        }
    }
```

Para obtener una lista completa de las guías para desarrolladores del AWS SDK y ejemplos de código, consulte Uso de este servicio con un SDK de AWS. En este tema también se incluye información sobre cómo comenzar a utilizar el SDK y detalles sobre sus versiones anteriores.

Ejecución de una copia multiparte de un objeto de Amazon S3 con un SDK de AWS

El siguiente ejemplo de código muestra cómo realizar una copia multiparte de un objeto de Amazon S3.

.NET

AWS SDK for .NET

> **Note**
>
> Hay más en GitHub. Busque el ejemplo completo y aprenda a configurar y ejecutar en el Repositorio de ejemplos de código de 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");
            }
        }
    }
```

- Para obtener información sobre la API, consulte los siguientes temas en la referencia de la API de AWS SDK for .NET.
  - CompleteMultipartUpload
  - CreateMultipartUpload
  - GetObjectMetadata
  - UploadPartCopy

Para obtener una lista completa de las guías para desarrolladores del AWS SDK y ejemplos de código, consulte Uso de este servicio con un SDK de AWS. En este tema también se incluye información sobre cómo comenzar a utilizar el SDK y detalles sobre sus versiones anteriores.

Ejecución de una carga multiparte de un objeto de Amazon S3 con un AWS SDK

En el siguiente ejemplo de código, se muestra cómo realizar una carga multiparte a un objeto de Amazon S3.

Java

SDK para Java 2.x

> **Note**
>
> Hay más en GitHub. Busque el ejemplo completo y aprenda a configurar y ejecutar en el Repositorio de ejemplos de código de AWS.

En los ejemplos de código se utilizan las siguientes importaciones.

```
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;
```

Utilice el Gestor de transferencias de Amazon S3 situado sobre el cliente S3 basado en CRT de AWS para realizar de forma transparente una carga multiparte cuando el tamaño del contenido supere un umbral. El umbral de tamaño predeterminado es 8 MB.

```
    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();
    }
```

Utilice la API S3Client para realizar una carga multiparte.

```
    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()));
    }
```

Utilice la API S3AsyncClient con soporte multiparte habilitado para realizar una carga multiparte.

```
    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.");
    }
```

- Para obtener información sobre la API, consulte los siguientes temas en la referencia de la API de AWS SDK for Java 2.x.
  - CompleteMultipartUpload
  - CreateMultipartUpload
  - UploadPart

Para obtener una lista completa de las guías para desarrolladores del AWS SDK y ejemplos de código, consulte Uso de este servicio con un SDK de AWS. En este tema también se incluye información sobre cómo comenzar a utilizar el SDK y detalles sobre sus versiones anteriores.

Reciba y procese las notificaciones de eventos de Amazon S3 mediante un AWS SDK.

El siguiente ejemplo de código muestra cómo trabajar con notificaciones de eventos de S3 de una forma orientada a objetos.

Java

SDK para Java 2.x

> **Note**
>
> Hay más en GitHub. Busque el ejemplo completo y aprenda a configurar y ejecutar en el Repositorio de ejemplos de código de AWS.

Este ejemplo muestra cómo procesar un evento de notificación de S3 utilizando Amazon SQS.

```
    /**
     * This method receives S3 event notifications by using an SqsAsyncClient.
     * After the client receives the messages it deserializes the JSON payload
and logs them. It uses
     * the S3EventNotification class (part of the S3 event notification API for
Java) to deserialize
     * the JSON payload and access the messages in an object-oriented way.
     *
     * @param queueUrl The URL of the AWS SQS queue that receives the S3 event
notifications.
```

```
    * @see <a href="https://sdk.amazonaws.com/java/api/latest/software/amazon/
awssdk/eventnotifications/s3/model/package-summary.html">S3EventNotification
 API</a>.
    * <p>
    * To use S3 event notification serialization/deserialization to objects, add
 the following
    * dependency to your Maven pom.xml file.
    * <dependency>
    * <groupId>software.amazon.awssdk</groupId>
    * <artifactId>s3-event-notifications</artifactId>
    * <version><LATEST></version>
    * </dependency>
    * <p>
    * The S3 event notification API became available with version 2.25.11 of the
 Java SDK.
    * <p>
    * This example shows the use of the API with AWS SQS, but it can be used to
 process S3 event notifications
    * in AWS SNS or AWS Lambda as well.
    * <p>
    * Note: The S3EventNotification class does not work with messages routed
 through AWS EventBridge.
    */
   static void processS3Events(String bucketName, String queueUrl, String
 queueArn) {
       try {
           // Configure the bucket to send Object Created and Object Tagging
 notifications to an existing SQS queue.
           s3Client.putBucketNotificationConfiguration(b -> b
                   .notificationConfiguration(ncb -> ncb
                           .queueConfigurations(qcb -> qcb
                                   .events(Event.S3_OBJECT_CREATED,
 Event.S3_OBJECT_TAGGING)
                                   .queueArn(queueArn)))
                           .bucket(bucketName)
           ).join();

           triggerS3EventNotifications(bucketName);
           // Wait for event notifications to propagate.
           Thread.sleep(Duration.ofSeconds(5).toMillis());

           boolean didReceiveMessages = true;
           while (didReceiveMessages) {
```

```
                // Display the number of messages that are available in the
queue.
                sqsClient.getQueueAttributes(b -> b
                                .queueUrl(queueUrl)

.attributeNames(QueueAttributeName.APPROXIMATE_NUMBER_OF_MESSAGES)
                        ).thenAccept(attributeResponse ->
                                logger.info("Approximate number of messages in
the queue: {}",

attributeResponse.attributes().get(QueueAttributeName.APPROXIMATE_NUMBER_OF_MESSAGES)))
                        .join();

                // Receive the messages.
                ReceiveMessageResponse response = sqsClient.receiveMessage(b -> b
                        .queueUrl(queueUrl)
                ).get();
                logger.info("Count of received messages: {}",
response.messages().size());
                didReceiveMessages = !response.messages().isEmpty();

                // Create a collection to hold the received message for deletion
                // after we log the messages.
                HashSet<DeleteMessageBatchRequestEntry> messagesToDelete = new
HashSet<>();
                // Process each message.
                response.messages().forEach(message -> {
                    logger.info("Message id: {}", message.messageId());
                    // Deserialize JSON message body to a S3EventNotification
object
                    // to access messages in an object-oriented way.
                    S3EventNotification event =
S3EventNotification.fromJson(message.body());

                    // Log the S3 event notification record details.
                    if (event.getRecords() != null) {
                        event.getRecords().forEach(record -> {
                            String eventName = record.getEventName();
                            String key = record.getS3().getObject().getKey();
                            logger.info(record.toString());
                            logger.info("Event name is {} and key is {}",
eventName, key);
                        });
                    }
```

```
                    // Add logged messages to collection for batch deletion.
                    messagesToDelete.add(DeleteMessageBatchRequestEntry.builder()
                            .id(message.messageId())
                            .receiptHandle(message.receiptHandle())
                            .build());
                });
                // Delete messages.
                if (!messagesToDelete.isEmpty()) {

sqsClient.deleteMessageBatch(DeleteMessageBatchRequest.builder()
                            .queueUrl(queueUrl)
                            .entries(messagesToDelete)
                            .build()
                    ).join();
                }
            } // End of while block.
        } catch (InterruptedException | ExecutionException e) {
            throw new RuntimeException(e);
        }
    }
```

- Para obtener información sobre la API, consulte los siguientes temas en la referencia de la API de AWS SDK for Java 2.x.
  - DeleteMessageBatch
  - GetQueueAttributes
  - PutBucketNotificationConfiguration
  - ReceiveMessage

Para obtener una lista completa de las guías para desarrolladores del AWS SDK y ejemplos de código, consulte Uso de este servicio con un SDK de AWS. En este tema también se incluye información sobre cómo comenzar a utilizar el SDK y detalles sobre sus versiones anteriores.

Envío de notificaciones de eventos de S3 a Amazon EventBridge mediante un AWS SDK

El siguiente ejemplo de código muestra cómo habilitar un bucket para enviar notificaciones de eventos de S3 a EventBridge y enrutar las notificaciones a un tema de Amazon SNS y a una cola de Amazon SQS.

Java

SDK para Java 2.x

> **Note**
>
> Hay más en GitHub. Busque el ejemplo completo y aprenda a configurar y ejecutar en el Repositorio de ejemplos de código de AWS.

```
   /** This method configures a bucket to send events to AWS EventBridge and
creates a rule
    * to route the S3 object created events to a topic and a queue.
    *
    * @param bucketName Name of existing bucket
    * @param topicArn ARN of existing topic to receive S3 event notifications
    * @param queueArn ARN of existing queue to receive S3 event notifications
    *
    *  An AWS CloudFormation stack sets up the bucket, queue, topic before the
method runs.
    */
   public static String setBucketNotificationToEventBridge(String bucketName,
String topicArn, String queueArn) {
       try {
           // Enable bucket to emit S3 Event notifications to EventBridge.
           s3Client.putBucketNotificationConfiguration(b -> b
                   .bucket(bucketName)
                   .notificationConfiguration(b1 -> b1
                           .eventBridgeConfiguration(
                                   SdkBuilder::build)
                   ).build()).join();

           // Create an EventBridge rule to route Object Created notifications.
           PutRuleRequest putRuleRequest = PutRuleRequest.builder()
```

```
                    .name(RULE_NAME)
                    .eventPattern("""
                            {
                              "source": ["aws.s3"],
                              "detail-type": ["Object Created"],
                              "detail": {
                                "bucket": {
                                  "name": ["%s"]
                                }
                              }
                            }
                            """.formatted(bucketName))
                    .build();

            // Add the rule to the default event bus.
            PutRuleResponse putRuleResponse =
eventBridgeClient.putRule(putRuleRequest)
                    .whenComplete((r, t) -> {
                        if (t != null) {
                            logger.error("Error creating event bus rule: " +
t.getMessage(), t);
                            throw new RuntimeException(t.getCause().getMessage(),
t);
                        }
                        logger.info("Event bus rule creation request sent
successfully. ARN is: {}", r.ruleArn());
                    }).join();

            // Add the existing SNS topic and SQS queue as targets to the rule.
            eventBridgeClient.putTargets(b -> b
                    .eventBusName("default")
                    .rule(RULE_NAME)
                    .targets(List.of (
                            Target.builder()
                                    .arn(queueArn)
                                    .id("Queue")
                                    .build(),
                            Target.builder()
                                    .arn(topicArn)
                                    .id("Topic")
                                    .build())
                            )
                    ).join();
            return putRuleResponse.ruleArn();
```

```
        } catch (S3Exception e) {
            System.err.println(e.awsErrorDetails().errorMessage());
            System.exit(1);
        }
        return null;
    }
```

- Para obtener información sobre la API, consulte los siguientes temas en la referencia de la API de AWS SDK for Java 2.x.
  - PutBucketNotificationConfiguration
  - PutRule
  - PutTargets

Para obtener una lista completa de las guías para desarrolladores del AWS SDK y ejemplos de código, consulte Uso de este servicio con un SDK de AWS. En este tema también se incluye información sobre cómo comenzar a utilizar el SDK y detalles sobre sus versiones anteriores.

Realización de un seguimiento de la carga o descarga de un objeto de Amazon S3 mediante un AWS SDK

En el siguiente ejemplo de código se muestra cómo realizar un seguimiento de cargas o descargas de objetos de Amazon S3.

Java

SDK para Java 2.x

> **Note**
>
> Hay más en GitHub. Busque el ejemplo completo y aprenda a configurar y ejecutar en el Repositorio de ejemplos de código de AWS.

Realice el seguimiento del progreso de una carga de archivos.

```
    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!
        */
    }
```

Realice el seguimiento del progreso de una descarga de archivos.

```
   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!
        */
    }
```

- Para obtener información sobre la API, consulte los siguientes temas en la referencia de la API de AWS SDK for Java 2.x.
  - GetObject
  - PutObject

Para obtener una lista completa de las guías para desarrolladores del AWS SDK y ejemplos de código, consulte Uso de este servicio con un SDK de AWS. En este tema también se incluye información sobre cómo comenzar a utilizar el SDK y detalles sobre sus versiones anteriores.

Ejemplos de enfoques para pruebas unitarias y de integración con un SDK de AWS

El siguiente ejemplo de código muestra ejemplos de técnicas de prácticas recomendadas a la hora de escribir pruebas unitarias y de integración mediante un SDK de AWS.

Rust

SDK para Rust

> **Note**
>
> Hay más información en GitHub. Busque el ejemplo completo y aprenda a configurar y ejecutar en el Repositorio de ejemplos de código de AWS.

Cargo.toml para ver ejemplos de pruebas.

```
[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"
```

Ejemplo de pruebas unitarias con automock y un encapsulador de servicios.

```
// 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]
```

Ejemplo de pruebas de integración con 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]
```

Para obtener una lista completa de las guías para desarrolladores del AWS SDK y ejemplos de código, consulte Uso de este servicio con un SDK de AWS. En este tema también se incluye información sobre cómo comenzar a utilizar el SDK y detalles sobre sus versiones anteriores.

Cargar de forma recursiva un directorio local en un bucket de Amazon Simple Storage Service (Amazon S3)

El siguiente ejemplo de código muestra cómo cargar de forma recursiva un directorio local en un bucket de Amazon Simple Storage Service (Amazon S3).

Java

SDK para Java 2.x

> **Note**
>
> Hay más en GitHub. Busque el ejemplo completo y aprenda a configurar y ejecutar en el Repositorio de ejemplos de código de AWS.

Utilice un S3TransferManager para cargar un directorio local. Vea el archivo completo y pruébelo.

```
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();
    }
```

- Para obtener información acerca de la API, consulte UploadDirectory en la Referencia de la API de AWS SDK for Java 2.x.

Para obtener una lista completa de las guías para desarrolladores del AWS SDK y ejemplos de código, consulte Uso de este servicio con un SDK de AWS. En este tema también se incluye información sobre cómo comenzar a utilizar el SDK y detalles sobre sus versiones anteriores.

Cargar o descargar archivos de gran tamaño desde y hacia Amazon S3 con un SDK de AWS

En los siguientes ejemplos de código se muestra cómo cargar o descargar archivos grandes hacia y desde Amazon S3.

Para obtener información, consulte Carga de un objeto con carga multiparte.

.NET

AWS SDK for .NET

> **Note**
>
> Hay más información en GitHub. Busque el ejemplo completo y aprenda a configurar y ejecutar en el Repositorio de ejemplos de código de AWS.

Llame a funciones que transfieren archivos hacia y desde un bucket de S3 mediante 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);
}
```

Cargar un solo archivo.

```
        /// <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;
            }
        }
```

Cargar un directorio local completo.

```
        /// <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;
            }
        }
```

Descargar un solo archivo.

```
        /// <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}"));
        }
```

Descargar el contenido de un bucket de 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;
        }
```

Realice un seguimiento del progreso de una carga mediante la 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}");
        }
    }
```

Cargar un objeto con cifrado.

```
    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

SDK para Go V2

> **Note**
>
> Hay más en GitHub. Busque el ejemplo completo y aprenda a configurar y ejecutar en el Repositorio de ejemplos de código de AWS.

Cargue un objeto grande mediante un mánager de carga para dividir los datos en partes y cargarlos simultáneamente.

```
// 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
}
```

Descargue un objeto grande mediante un mánager de descargas para obtener los datos en partes y descargarlos simultáneamente.

```
// 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 para Java 2.x

> **Note**
>
> Hay más en GitHub. Busque el ejemplo completo y aprenda a configurar y ejecutar en el Repositorio de ejemplos de código de AWS.

Llame a funciones que transfieran archivos hacia y desde un bucket de S3 mediante 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();
    }
```

Cargar un directorio local completo.

```
   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();
   }
```

Cargar un solo archivo.

```
   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 para JavaScript (v3)

> **Note**
>
> Hay más información en GitHub. Busque el ejemplo completo y aprenda a configurar y ejecutar en el Repositorio de ejemplos de código de AWS.

Cargue un archivo grande.

```
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);
    }
  }
};
```

Descargue un archivo grande.

```
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 para Python (Boto3)

> **Note**
>
> Hay más en GitHub. Busque el ejemplo completo y aprenda a configurar y ejecutar en el Repositorio de ejemplos de código de AWS.

Cree funciones que transfieran archivos utilizando varios de los ajustes del administrador de transferencias disponibles. Utilice una clase de devolución de llamada para escribir el progreso de la devolución de llamada durante la transferencia de archivos.

```
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
```

Demuestre las funciones del administrador de transferencias e informe de los resultados.

```
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 para Rust

> **Note**
>
> Hay más información en GitHub. Busque el ejemplo completo y aprenda a configurar y ejecutar en el Repositorio de ejemplos de código de 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(())
}
```

Para obtener una lista completa de las guías para desarrolladores del AWS SDK y ejemplos de código, consulte Uso de este servicio con un SDK de AWS. En este tema también se incluye información sobre cómo comenzar a utilizar el SDK y detalles sobre sus versiones anteriores.

Carga de un flujo de tamaño desconocido en un objeto de Amazon S3 mediante un SDK de AWS

En los siguientes ejemplos de código, se muestra cómo cargar un flujo de tamaño desconocido en un objeto de Amazon S3.

Java

SDK para Java 2.x

> **Note**
>
> Hay más en GitHub. Busque el ejemplo completo y aprenda a configurar y ejecutar en el Repositorio de ejemplos de código de AWS.

Utilice el Cliente S3 basado en CRT de 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;
    }
```

```
}
```

Utilice el Gestor de transferencias de 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();
    }
}
```

Para obtener una lista completa de las guías para desarrolladores del AWS SDK y ejemplos de código, consulte Uso de este servicio con un SDK de AWS. En este tema también se incluye información sobre cómo comenzar a utilizar el SDK y detalles sobre sus versiones anteriores.

Uso de sumas de comprobación para trabajar con un objeto de Amazon S3 con un SDK de AWS

En el siguiente ejemplo de código, se muestra cómo utilizar sumas de comprobación para trabajar con un objeto de Amazon S3.

Java

SDK para Java 2.x

> **Note**
>
> Hay más en GitHub. Busque el ejemplo completo y aprenda a configurar y ejecutar en el Repositorio de ejemplos de código de AWS.

En los ejemplos de código se utiliza un subconjunto de las siguientes importaciones.

```
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;
```

Especifique un algoritmo de suma de comprobación para el método `putObject` al crear la `PutObjectRequest`.

```
    public void putObjectWithChecksum() {
        s3Client.putObject(b -> b
                .bucket(bucketName)
                .key(key)
                .checksumAlgorithm(ChecksumAlgorithm.CRC32),
                RequestBody.fromString("This is a test"));
    }
```

Compruebe la suma de comprobación del método `getObject` cuando cree la `GetObjectRequest`.

```
    public GetObjectResponse getObjectWithChecksum() {
        return s3Client.getObject(b -> b
                .bucket(bucketName)
                .key(key)
                .checksumMode(ChecksumMode.ENABLED))
                .response();
    }
```

Calcule previamente una suma de comprobación para el método `putObject` cuando cree la `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)));
    }
```

Utilice el Gestor de transferencias de Amazon S3 situado sobre el cliente S3 basado en CRT de AWS para realizar de forma transparente una carga multiparte cuando el tamaño del contenido supere un umbral. El umbral de tamaño predeterminado es 8 MB.

Puede especificar un algoritmo de suma de comprobación para que lo utilice el SDK. De forma predeterminada, el SDK usa el algoritmo 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();
    }
```

Utilice la API S3Client o (la API S3AsyncClient) para realizar una carga multiparte. Si especifica una suma de comprobación adicional, debe especificar el algoritmo que se utilizará al iniciar la carga. También debe especificar el algoritmo para cada solicitud de parte y proporcionar la suma de comprobación calculada para cada parte una vez cargada.

```
    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()));
    }
```

- Para obtener información sobre la API, consulte los siguientes temas en la referencia de la API de AWS SDK for Java 2.x.
  - CompleteMultipartUpload
  - CreateMultipartUpload
  - UploadPart

Para obtener una lista completa de las guías para desarrolladores del AWS SDK y ejemplos de código, consulte Uso de este servicio con un SDK de AWS. En este tema también se incluye información sobre cómo comenzar a utilizar el SDK y detalles sobre sus versiones anteriores.

Trabajo con las características de integridad de objetos de Amazon S3 utilizando un AWS SDK

El siguiente ejemplo de código muestra cómo trabajar con las características de integridad de objetos de S3.

C++

SDK para C++

> **Note**
>
> Hay más en GitHub. Busque el ejemplo completo y aprenda a configurar y ejecutar en el Repositorio de ejemplos de código de AWS.

Ejecute un escenario interactivo en el que se demuestren las características de integridad de objetos de Amazon S3.

```
//! Routine which runs the S3 object integrity workflow.
/*!
   \param clientConfig: Aws client configuration.
   \return bool: Function succeeded.
*/
bool AwsDoc::S3::s3ObjectIntegrityWorkflow(
        const Aws::S3::S3ClientConfiguration &clientConfiguration) {

    /*
     * Create a large file to be used for multipart uploads.
     */
    if (!createLargeFileIfNotExists()) {
        std::cerr << "Workflow exiting because large file creation failed." <<
 std::endl;
        return false;
    }

    Aws::String bucketName = TEST_BUCKET_PREFIX;
    bucketName += Aws::Utils::UUID::RandomUUID();
    bucketName = Aws::Utils::StringUtils::ToLower(bucketName.c_str());

    bucketName.resize(std::min(bucketName.size(), MAX_BUCKET_NAME_LENGTH));
```

```
   introductoryExplanations(bucketName);

   if (!AwsDoc::S3::createBucket(bucketName, clientConfiguration)) {
       std::cerr << "Workflow exiting because bucket creation failed." <<
std::endl;
       return false;
   }

   Aws::S3::S3ClientConfiguration s3ClientConfiguration(clientConfiguration);
   std::shared_ptr<Aws::S3::S3Client> client =
Aws::MakeShared<Aws::S3::S3Client>("S3Client", s3ClientConfiguration);

   printAsterisksLine();
   std::cout << "Choose from one of the following checksum algorithms."
             << std::endl;

   for (HASH_METHOD hashMethod = DEFAULT; hashMethod <= SHA256; ++hashMethod) {
       std::cout << "  " << hashMethod << " - " <<
stringForHashMethod(hashMethod)
                 << std::endl;
   }

   HASH_METHOD chosenHashMethod = askQuestionForIntRange("Enter an index: ",
DEFAULT,
                                                         SHA256);


   gUseCalculatedChecksum = !askYesNoQuestion(
           "Let the SDK calculate the checksum for you? (y/n) ");

   printAsterisksLine();

   std::cout << "The workflow will now upload a file using PutObject."
             << std::endl;
   std::cout << "Object integrity will be verified using the "
             << stringForHashMethod(chosenHashMethod) << " algorithm."
             << std::endl;
   if (gUseCalculatedChecksum) {
       std::cout
               << "A checksum computed by this workflow will be used for object
integrity verification,"
               << std::endl;
       std::cout << "except for the TransferManager upload." << std::endl;
```

```
    } else {
        std::cout
                << "A checksum computed by the SDK will be used for object
integrity verification."
                << std::endl;
    }

    pressEnterToContinue();
    printAsterisksLine();

    std::shared_ptr<Aws::IOStream> inputData =
            Aws::MakeShared<Aws::FStream>("SampleAllocationTag",
                                          TEST_FILE,
                                          std::ios_base::in |
                                          std::ios_base::binary);

    if (!*inputData) {
        std::cerr << "Error unable to read file " << TEST_FILE << std::endl;
        cleanUp(bucketName, clientConfiguration);
        return false;
    }

    Hasher hasher;
    HASH_METHOD putObjectHashMethod = chosenHashMethod;
    if (putObjectHashMethod == DEFAULT) {
        putObjectHashMethod = MD5; // MD5 is the default hash method for
PutObject.

        std::cout << "The default checksum algorithm for PutObject is "
                  << stringForHashMethod(putObjectHashMethod)
                  << std::endl;
    }

    // Demonstrate in code how the hash is computed.
    if (!hasher.calculateObjectHash(*inputData, putObjectHashMethod)) {
        std::cerr << "Error calculating hash for file " << TEST_FILE <<
std::endl;
        cleanUp(bucketName, clientConfiguration);
        return false;
    }
    Aws::String key = stringForHashMethod(putObjectHashMethod);
    key += "_";
    key += TEST_FILE_KEY;
    Aws::String localHash = hasher.getBase64HashString();
```

```
    // Upload the object with PutObject
    if (!putObjectWithHash(bucketName, key, localHash, putObjectHashMethod,
                           inputData, chosenHashMethod == DEFAULT,
                           *client)) {
        std::cerr << "Error putting file " << TEST_FILE << " to bucket "
                  << bucketName << " with key " << key << std::endl;
        cleanUp(bucketName, clientConfiguration);
        return false;
    }

    Aws::String retrievedHash;
    if (!retrieveObjectHash(bucketName, key,
                            putObjectHashMethod, retrievedHash,
                            nullptr, *client)) {
        std::cerr << "Error getting file " << TEST_FILE << " from bucket "
                  << bucketName << " with key " << key << std::endl;
        cleanUp(bucketName, clientConfiguration);
        return false;
    }

    explainPutObjectResults();
    verifyHashingResults(retrievedHash, hasher,
                         "PutObject upload", putObjectHashMethod);


    printAsterisksLine();
    pressEnterToContinue();

    key = "tr_";
    key += stringForHashMethod(chosenHashMethod) + "_" + MULTI_PART_TEST_FILE;

    introductoryTransferManagerUploadExplanations(key);

    HASH_METHOD transferManagerHashMethod = chosenHashMethod;
    if (transferManagerHashMethod == DEFAULT) {
        transferManagerHashMethod = CRC32;  // The default hash method for the
TransferManager is CRC32.

        std::cout << "The default checksum algorithm for TransferManager is "
                  << stringForHashMethod(transferManagerHashMethod)
                  << std::endl;
    }
```

```
   // Upload the large file using the transfer manager.
   if (!doTransferManagerUpload(bucketName, key, transferManagerHashMethod,
chosenHashMethod == DEFAULT,
                                client)) {
       std::cerr << "Exiting because of an error in doTransferManagerUpload." <<
std::endl;
       cleanUp(bucketName, clientConfiguration);
       return false;
   }

   std::vector<Aws::String> retrievedTransferManagerPartHashes;
   Aws::String retrievedTransferManagerFinalHash;

   // Retrieve all the hashes for the TransferManager upload.
   if (!retrieveObjectHash(bucketName, key,
                           transferManagerHashMethod,
                           retrievedTransferManagerFinalHash,
                           &retrievedTransferManagerPartHashes, *client)) {
       std::cerr << "Exiting because of an error in retrieveObjectHash for
TransferManager." << std::endl;
       cleanUp(bucketName, clientConfiguration);
       return false;
   }

   AwsDoc::S3::Hasher locallyCalculatedFinalHash;
   std::vector<Aws::String> locallyCalculatedPartHashes;

   // Calculate the hashes locally to demonstrate how TransferManager hashes are
computed.
   if (!calculatePartHashesForFile(transferManagerHashMethod,
MULTI_PART_TEST_FILE,
                                   UPLOAD_BUFFER_SIZE,
                                   locallyCalculatedFinalHash,
                                   locallyCalculatedPartHashes)) {
       std::cerr << "Exiting because of an error in calculatePartHashesForFile."
<< std::endl;
       cleanUp(bucketName, clientConfiguration);
       return false;
   }

   verifyHashingResults(retrievedTransferManagerFinalHash,
                        locallyCalculatedFinalHash, "TransferManager upload",
                        transferManagerHashMethod,
                        retrievedTransferManagerPartHashes,
```

```
                        locallyCalculatedPartHashes);

    printAsterisksLine();

    key = "mp_";
    key += stringForHashMethod(chosenHashMethod) + "_" + MULTI_PART_TEST_FILE;

    multiPartUploadExplanations(key, chosenHashMethod);

    pressEnterToContinue();

    std::shared_ptr<Aws::IOStream> largeFileInputData =
            Aws::MakeShared<Aws::FStream>("SampleAllocationTag",
                                          MULTI_PART_TEST_FILE,
                                          std::ios_base::in |
                                          std::ios_base::binary);

    if (!largeFileInputData->good()) {
        std::cerr << "Error unable to read file " << TEST_FILE << std::endl;
        cleanUp(bucketName, clientConfiguration);
        return false;
    }

    HASH_METHOD multipartUploadHashMethod = chosenHashMethod;
    if (multipartUploadHashMethod == DEFAULT) {
        multipartUploadHashMethod = MD5;  // The default hash method for
multipart uploads is MD5.

        std::cout << "The default checksum algorithm for multipart upload is "
                  << stringForHashMethod(putObjectHashMethod)
                  << std::endl;
    }

    AwsDoc::S3::Hasher hashData;
    std::vector<Aws::String> partHashes;

    if (!doMultipartUpload(bucketName, key,
                           multipartUploadHashMethod,
                           largeFileInputData, chosenHashMethod == DEFAULT,
                           hashData,
                           partHashes,
                           *client)) {
        std::cerr << "Exiting because of an error in doMultipartUpload." <<
std::endl;
```

```
        cleanUp(bucketName, clientConfiguration);
        return false;
    }

    std::cout << "Finished multipart upload of with hash method " <<
              stringForHashMethod(multipartUploadHashMethod) << std::endl;

    std::cout << "Now we will retrieve the checksums from the server." <<
 std::endl;

    retrievedHash.clear();
    std::vector<Aws::String> retrievedPartHashes;
    if (!retrieveObjectHash(bucketName, key,
                            multipartUploadHashMethod,
                            retrievedHash, &retrievedPartHashes, *client)) {
        std::cerr << "Exiting because of an error in retrieveObjectHash for
 multipart." << std::endl;
        cleanUp(bucketName, clientConfiguration);
        return false;
    }

    verifyHashingResults(retrievedHash, hashData, "MultiPart upload",
                         multipartUploadHashMethod,
                         retrievedPartHashes, partHashes);

    printAsterisksLine();

    if (askYesNoQuestion("Would you like to delete the resources created in this
 workflow? (y/n)")) {
        return cleanUp(bucketName, clientConfiguration);
    } else {
        std::cout << "The bucket " << bucketName << " was not deleted." <<
 std::endl;
        return true;
    }
}

//! Routine which uploads an object to an S3 bucket with different object
 integrity hashing methods.
/*!
   \param bucket: The name of the S3 bucket where the object will be uploaded.
   \param key: The unique identifier (key) for the object within the S3 bucket.
   \param hashData: The hash value that will be associated with the uploaded
 object.
```

```
   \param hashMethod: The hashing algorithm to use when calculating the hash
 value.
   \param body: The data content of the object being uploaded.
   \param useDefaultHashMethod: A flag indicating whether to use the default hash
 method or the one specified in the hashMethod parameter.
   \param client: The S3 client instance used to perform the upload operation.
   \return bool: Function succeeded.
*/
bool AwsDoc::S3::putObjectWithHash(const Aws::String &bucket, const Aws::String
 &key,
                                   const Aws::String &hashData,
                                   AwsDoc::S3::HASH_METHOD hashMethod,
                                   const std::shared_ptr<Aws::IOStream> &body,
                                   bool useDefaultHashMethod,
                                   const Aws::S3::S3Client &client) {
   Aws::S3::Model::PutObjectRequest request;
   request.SetBucket(bucket);
   request.SetKey(key);
   if (!useDefaultHashMethod) {
       if (hashMethod != MD5) {

 request.SetChecksumAlgorithm(getChecksumAlgorithmForHashMethod(hashMethod));
       }
   }

   if (gUseCalculatedChecksum) {
       switch (hashMethod) {
           case AwsDoc::S3::MD5:
               request.SetContentMD5(hashData);
               break;
           case AwsDoc::S3::SHA1:
               request.SetChecksumSHA1(hashData);
               break;
           case AwsDoc::S3::SHA256:
               request.SetChecksumSHA256(hashData);
               break;
           case AwsDoc::S3::CRC32:
               request.SetChecksumCRC32(hashData);
               break;
           case AwsDoc::S3::CRC32C:
               request.SetChecksumCRC32C(hashData);
               break;
           default:
               std::cerr << "Unknown hash method." << std::endl;
```

```
                return false;
        }
    }
    request.SetBody(body);
    Aws::S3::Model::PutObjectOutcome outcome = client.PutObject(request);
    body->seekg(0, body->beg);
    if (outcome.IsSuccess()) {
        std::cout << "Object successfully uploaded." << std::endl;
    } else {
        std::cerr << "Error uploading object." <<
                  outcome.GetError().GetMessage() << std::endl;
    }
    return outcome.IsSuccess();
}


// ! Routine which retrieves the hash value of an object stored in an S3 bucket.
/*!
   \param bucket: The name of the S3 bucket where the object is stored.
   \param key: The unique identifier (key) of the object within the S3 bucket.
   \param hashMethod: The hashing algorithm used to calculate the hash value of
 the object.
   \param[out] hashData: The retrieved hash.
   \param[out] partHashes: The part hashes if available.
   \param client: The S3 client instance used to retrieve the object.
   \return bool: Function succeeded.
*/
bool AwsDoc::S3::retrieveObjectHash(const Aws::String &bucket, const Aws::String
 &key,
                                    AwsDoc::S3::HASH_METHOD hashMethod,
                                    Aws::String &hashData,
                                    std::vector<Aws::String> *partHashes,
                                    const Aws::S3::S3Client &client) {
    Aws::S3::Model::GetObjectAttributesRequest request;
    request.SetBucket(bucket);
    request.SetKey(key);

    if (hashMethod == MD5) {
        Aws::Vector<Aws::S3::Model::ObjectAttributes> attributes;
        attributes.push_back(Aws::S3::Model::ObjectAttributes::ETag);
        request.SetObjectAttributes(attributes);

        Aws::S3::Model::GetObjectAttributesOutcome outcome =
 client.GetObjectAttributes(
```

```
                request);
        if (outcome.IsSuccess()) {
            const Aws::S3::Model::GetObjectAttributesResult &result =
 outcome.GetResult();
            hashData = result.GetETag();
        } else {
            std::cerr << "Error retrieving object etag attributes." <<
                      outcome.GetError().GetMessage() << std::endl;
            return false;
        }
    } else { // hashMethod != MD5
        Aws::Vector<Aws::S3::Model::ObjectAttributes> attributes;
        attributes.push_back(Aws::S3::Model::ObjectAttributes::Checksum);
        request.SetObjectAttributes(attributes);

        Aws::S3::Model::GetObjectAttributesOutcome outcome =
 client.GetObjectAttributes(
                request);
        if (outcome.IsSuccess()) {
            const Aws::S3::Model::GetObjectAttributesResult &result =
 outcome.GetResult();
            switch (hashMethod) {
                case AwsDoc::S3::DEFAULT: // NOLINT(*-branch-clone)
                    break;  // Default is not supported.
#pragma clang diagnostic push
#pragma ide diagnostic ignored "UnreachableCode"
                case AwsDoc::S3::MD5:
                    break;  // MD5 is not supported.
#pragma clang diagnostic pop
                case AwsDoc::S3::SHA1:
                    hashData = result.GetChecksum().GetChecksumSHA1();
                    break;
                case AwsDoc::S3::SHA256:
                    hashData = result.GetChecksum().GetChecksumSHA256();
                    break;
                case AwsDoc::S3::CRC32:
                    hashData = result.GetChecksum().GetChecksumCRC32();
                    break;
                case AwsDoc::S3::CRC32C:
                    hashData = result.GetChecksum().GetChecksumCRC32C();
                    break;
                default:
                    std::cerr << "Unknown hash method." << std::endl;
                    return false;
```

```
            }
        } else {
            std::cerr << "Error retrieving object checksum attributes." <<
                      outcome.GetError().GetMessage() << std::endl;
            return false;
        }

        if (nullptr != partHashes) {
            attributes.clear();
            attributes.push_back(Aws::S3::Model::ObjectAttributes::ObjectParts);
            request.SetObjectAttributes(attributes);
            outcome = client.GetObjectAttributes(request);
            if (outcome.IsSuccess()) {
                const Aws::S3::Model::GetObjectAttributesResult &result =
outcome.GetResult();
                const Aws::Vector<Aws::S3::Model::ObjectPart> parts =
result.GetObjectParts().GetParts();
                for (const Aws::S3::Model::ObjectPart &part: parts) {
                    switch (hashMethod) {
                        case AwsDoc::S3::DEFAULT: // Default is not supported.
NOLINT(*-branch-clone)
                            break;
                        case AwsDoc::S3::MD5: // MD5 is not supported.
                            break;
                        case AwsDoc::S3::SHA1:
                            partHashes->push_back(part.GetChecksumSHA1());
                            break;
                        case AwsDoc::S3::SHA256:
                            partHashes->push_back(part.GetChecksumSHA256());
                            break;
                        case AwsDoc::S3::CRC32:
                            partHashes->push_back(part.GetChecksumCRC32());
                            break;
                        case AwsDoc::S3::CRC32C:
                            partHashes->push_back(part.GetChecksumCRC32C());
                            break;
                        default:
                            std::cerr << "Unknown hash method." << std::endl;
                            return false;
                    }
                }
            } else {
                std::cerr << "Error retrieving object attributes for object
parts." <<
```

```
                          outcome.GetError().GetMessage() << std::endl;
                return false;
            }
        }
    }

    return true;
}

//! Verifies the hashing results between the retrieved and local hashes.
/*!
 \param retrievedHash The hash value retrieved from the remote source.
 \param localHash The hash value calculated locally.
 \param uploadtype The type of upload (e.g., "multipart", "single-part").
 \param hashMethod The hashing method used (e.g., MD5, SHA-256).
 \param retrievedPartHashes (Optional) The list of hashes for the individual
 parts retrieved from the remote source.
 \param localPartHashes (Optional) The list of hashes for the individual parts
 calculated locally.
 */
void AwsDoc::S3::verifyHashingResults(const Aws::String &retrievedHash,
                                      const Hasher &localHash,
                                      const Aws::String &uploadtype,
                                      HASH_METHOD hashMethod,
                                      const std::vector<Aws::String>
 &retrievedPartHashes,
                                      const std::vector<Aws::String>
 &localPartHashes) {
    std::cout << "For " << uploadtype << " retrieved hash is " << retrievedHash
 << std::endl;
    if (!retrievedPartHashes.empty()) {
        std::cout << retrievedPartHashes.size() << " part hash(es) were also
 retrieved."
                  << std::endl;
        for (auto &retrievedPartHash: retrievedPartHashes) {
            std::cout << "  Part hash " << retrievedPartHash << std::endl;
        }
    }
    Aws::String hashString;
    if (hashMethod == MD5) {
        hashString = localHash.getHexHashString();
        if (!localPartHashes.empty()) {
            hashString += "-" + std::to_string(localPartHashes.size());
        }
```

```
   } else {
       hashString = localHash.getBase64HashString();
   }

   bool allMatch = true;
   if (hashString != retrievedHash) {
       std::cerr << "For " << uploadtype << ", the main hashes do not match" <<
std::endl;
       std::cerr << "Local hash- '" << hashString << "'" << std::endl;
       std::cerr << "Remote hash - '" << retrievedHash << "'" << std::endl;
       allMatch = false;
   }

   if (hashMethod != MD5) {
       if (localPartHashes.size() != retrievedPartHashes.size()) {
           std::cerr << "For " << uploadtype << ", the number of part hashes do
not match" << std::endl;
           std::cerr << "Local number of hashes- '" << localPartHashes.size() <<
"'"
                     << std::endl;
           std::cerr << "Remote number of hashes - '"
                     << retrievedPartHashes.size()
                     << "'" << std::endl;
       }

       for (int i = 0; i < localPartHashes.size(); ++i) {
           if (localPartHashes[i] != retrievedPartHashes[i]) {
               std::cerr << "For " << uploadtype << ", the part hashes do not
match for part " << i + 1
                         << "." << std::endl;
               std::cerr << "Local hash- '" << localPartHashes[i] << "'"
                         << std::endl;
               std::cerr << "Remote hash - '" << retrievedPartHashes[i] << "'"
                         << std::endl;
               allMatch = false;
           }
       }
   }

   if (allMatch) {
       std::cout << "For " << uploadtype << ", locally and remotely calculated
hashes all match!" << std::endl;
   }
```

```
}

static void transferManagerErrorCallback(const Aws::Transfer::TransferManager *,
                                         const std::shared_ptr<const
 Aws::Transfer::TransferHandle> &,
                                         const
 Aws::Client::AWSError<Aws::S3::S3Errors> &err) {
    std::cerr << "Error during transfer: '" << err.GetMessage() << "'" <<
 std::endl;
}

static void transferManagerStatusCallback(const Aws::Transfer::TransferManager *,
                                          const std::shared_ptr<const
 Aws::Transfer::TransferHandle> &handle) {
    if (handle->GetStatus() == Aws::Transfer::TransferStatus::IN_PROGRESS) {
        std::cout << "Bytes transferred: " << handle->GetBytesTransferred() <<
 std::endl;
    }
}

//! Routine which uploads an object to an S3 bucket using the AWS C++ SDK's
 Transfer Manager.
/*!
   \param bucket: The name of the S3 bucket where the object will be uploaded.
   \param key: The unique identifier (key) for the object within the S3 bucket.
   \param hashMethod: The hashing algorithm to use when calculating the hash
 value.
   \param useDefaultHashMethod: A flag indicating whether to use the default hash
 method or the one specified in the hashMethod parameter.
   \param client: The S3 client instance used to perform the upload operation.
   \return bool: Function succeeded.
*/
bool
AwsDoc::S3::doTransferManagerUpload(const Aws::String &bucket, const Aws::String
 &key,
                                    AwsDoc::S3::HASH_METHOD hashMethod,
                                    bool useDefaultHashMethod,
                                    const std::shared_ptr<Aws::S3::S3Client>
 &client) {
    std::shared_ptr<Aws::Utils::Threading::PooledThreadExecutor> executor =
 Aws::MakeShared<Aws::Utils::Threading::PooledThreadExecutor>(
            "executor", 25);
    Aws::Transfer::TransferManagerConfiguration transfer_config(executor.get());
    transfer_config.s3Client = client;
```

```
    transfer_config.bufferSize = UPLOAD_BUFFER_SIZE;
    if (!useDefaultHashMethod) {
        if (hashMethod == MD5) {
            transfer_config.computeContentMD5 = true;
        } else {
            transfer_config.checksumAlgorithm =
 getChecksumAlgorithmForHashMethod(
                    hashMethod);
        }
    }
    transfer_config.errorCallback = transferManagerErrorCallback;
    transfer_config.transferStatusUpdatedCallback =
 transferManagerStatusCallback;

    std::shared_ptr<Aws::Transfer::TransferManager> transfer_manager =
 Aws::Transfer::TransferManager::Create(
            transfer_config);

    std::cout << "Uploading the file..." << std::endl;
    std::shared_ptr<Aws::Transfer::TransferHandle> uploadHandle =
 transfer_manager->UploadFile(MULTI_PART_TEST_FILE,

              bucket, key,

              "text/plain",

              Aws::Map<Aws::String, Aws::String>());
    uploadHandle->WaitUntilFinished();
    bool success =
            uploadHandle->GetStatus() ==
 Aws::Transfer::TransferStatus::COMPLETED;
    if (!success) {
        Aws::Client::AWSError<Aws::S3::S3Errors> err = uploadHandle-
>GetLastError();
        std::cerr << "File upload failed:  " << err.GetMessage() << std::endl;
    }

    return success;
}

//! Routine which calculates the hash values for each part of a file being
 uploaded to an S3 bucket.
/*!
```

```
   \param hashMethod: The hashing algorithm to use when calculating the hash
 values.
   \param fileName: The path to the file for which the part hashes will be
 calculated.
   \param bufferSize: The size of the buffer to use when reading the file.
   \param[out] hashDataResult: The Hasher object that will store the concatenated
 hash value.
   \param[out] partHashes: The vector that will store the calculated hash values
 for each part of the file.
   \return bool: Function succeeded.
*/
bool AwsDoc::S3::calculatePartHashesForFile(AwsDoc::S3::HASH_METHOD hashMethod,
                                            const Aws::String &fileName,
                                            size_t bufferSize,
                                            AwsDoc::S3::Hasher &hashDataResult,
                                            std::vector<Aws::String> &partHashes)
 {
    std::ifstream fileStream(fileName.c_str(), std::ifstream::binary);
    fileStream.seekg(0, std::ifstream::end);
    size_t objectSize = fileStream.tellg();
    fileStream.seekg(0, std::ifstream::beg);
    std::vector<unsigned char> totalHashBuffer;
    size_t uploadedBytes = 0;


    while (uploadedBytes < objectSize) {
        std::vector<unsigned char> buffer(bufferSize);
        std::streamsize bytesToRead =
 static_cast<std::streamsize>(std::min(buffer.size(), objectSize -
 uploadedBytes));
        fileStream.read((char *) buffer.data(), bytesToRead);
        Aws::Utils::Stream::PreallocatedStreamBuf
 preallocatedStreamBuf(buffer.data(),

 bytesToRead);
        std::shared_ptr<Aws::IOStream> body =
                Aws::MakeShared<Aws::IOStream>("SampleAllocationTag",
                                               &preallocatedStreamBuf);
        Hasher hasher;
        if (!hasher.calculateObjectHash(*body, hashMethod)) {
            std::cerr << "Error calculating hash." << std::endl;
            return false;
        }
        Aws::String base64HashString = hasher.getBase64HashString();
```

```
        partHashes.push_back(base64HashString);

        Aws::Utils::ByteBuffer hashBuffer = hasher.getByteBufferHash();

        totalHashBuffer.insert(totalHashBuffer.end(),
 hashBuffer.GetUnderlyingData(),
                               hashBuffer.GetUnderlyingData() +
 hashBuffer.GetLength());

        uploadedBytes += bytesToRead;
    }

    return hashDataResult.calculateObjectHash(totalHashBuffer, hashMethod);
}

//! Create a multipart upload.
/*!
    \param bucket: The name of the S3 bucket where the object will be uploaded.
    \param key: The unique identifier (key) for the object within the S3 bucket.
    \param client: The S3 client instance used to perform the upload operation.
    \return Aws::String: Upload ID or empty string if failed.
*/
Aws::String
AwsDoc::S3::createMultipartUpload(const Aws::String &bucket, const Aws::String
 &key,
                                  Aws::S3::Model::ChecksumAlgorithm
 checksumAlgorithm,
                                  const Aws::S3::S3Client &client) {
    Aws::S3::Model::CreateMultipartUploadRequest request;
    request.SetBucket(bucket);
    request.SetKey(key);

    if (checksumAlgorithm != Aws::S3::Model::ChecksumAlgorithm::NOT_SET) {
        request.SetChecksumAlgorithm(checksumAlgorithm);
    }

    Aws::S3::Model::CreateMultipartUploadOutcome outcome =
            client.CreateMultipartUpload(request);

    Aws::String uploadID;
    if (outcome.IsSuccess()) {
        uploadID = outcome.GetResult().GetUploadId();
    } else {
```

```
        std::cerr << "Error creating multipart upload: " <<
 outcome.GetError().GetMessage() << std::endl;
    }

    return uploadID;
}

//! Upload a part to an S3 bucket.
/*!
    \param bucket: The name of the S3 bucket where the object will be uploaded.
    \param key: The unique identifier (key) for the object within the S3 bucket.
    \param uploadID: An upload ID string.
    \param partNumber:
    \param checksumAlgorithm: Checksum algorithm, ignored when NOT_SET.
    \param calculatedHash: A data integrity hash to set, depending on the
 checksum algorithm,
                            ignored when it is an empty string.
    \param body: An shared_ptr IOStream of the data to be uploaded.
    \param client: The S3 client instance used to perform the upload operation.
    \return UploadPartOutcome: The outcome.
*/

Aws::S3::Model::UploadPartOutcome AwsDoc::S3::uploadPart(const Aws::String
 &bucket,
                                                         const Aws::String &key,
                                                         const Aws::String
 &uploadID,
                                                         int partNumber,

 Aws::S3::Model::ChecksumAlgorithm checksumAlgorithm,
                                                         const Aws::String
 &calculatedHash,
                                                         const
 std::shared_ptr<Aws::IOStream> &body,
                                                         const Aws::S3::S3Client
 &client) {
    Aws::S3::Model::UploadPartRequest request;
    request.SetBucket(bucket);
    request.SetKey(key);
    request.SetUploadId(uploadID);
    request.SetPartNumber(partNumber);
    if (checksumAlgorithm != Aws::S3::Model::ChecksumAlgorithm::NOT_SET) {
        request.SetChecksumAlgorithm(checksumAlgorithm);
    }
```

```
    request.SetBody(body);

    if (!calculatedHash.empty()) {
        switch (checksumAlgorithm) {
            case Aws::S3::Model::ChecksumAlgorithm::NOT_SET:
                request.SetContentMD5(calculatedHash);
                break;
            case Aws::S3::Model::ChecksumAlgorithm::CRC32:
                request.SetChecksumCRC32(calculatedHash);
                break;
            case Aws::S3::Model::ChecksumAlgorithm::CRC32C:
                request.SetChecksumCRC32C(calculatedHash);
                break;
            case Aws::S3::Model::ChecksumAlgorithm::SHA1:
                request.SetChecksumSHA1(calculatedHash);
                break;
            case Aws::S3::Model::ChecksumAlgorithm::SHA256:
                request.SetChecksumSHA256(calculatedHash);
                break;
        }
    }

    return client.UploadPart(request);
}

//! Abort a multipart upload to an S3 bucket.
/*!
    \param bucket: The name of the S3 bucket where the object will be uploaded.
    \param key: The unique identifier (key) for the object within the S3 bucket.
    \param uploadID: An upload ID string.
    \param client: The S3 client instance used to perform the upload operation.
    \return bool: Function succeeded.
*/

bool AwsDoc::S3::abortMultipartUpload(const Aws::String &bucket,
                                      const Aws::String &key,
                                      const Aws::String &uploadID,
                                      const Aws::S3::S3Client &client) {
    Aws::S3::Model::AbortMultipartUploadRequest request;
    request.SetBucket(bucket);
    request.SetKey(key);
    request.SetUploadId(uploadID);

    Aws::S3::Model::AbortMultipartUploadOutcome outcome =
```

```
            client.AbortMultipartUpload(request);

    if (outcome.IsSuccess()) {
        std::cout << "Multipart upload aborted." << std::endl;
    } else {
        std::cerr << "Error aborting multipart upload: " <<
 outcome.GetError().GetMessage() << std::endl;
    }

    return outcome.IsSuccess();
}

//! Complete a multipart upload to an S3 bucket.
/*!
    \param bucket: The name of the S3 bucket where the object will be uploaded.
    \param key: The unique identifier (key) for the object within the S3 bucket.
    \param uploadID: An upload ID string.
    \param parts: A vector of CompleteParts.
    \param client: The S3 client instance used to perform the upload operation.
    \return CompleteMultipartUploadOutcome: The request outcome.
*/
Aws::S3::Model::CompleteMultipartUploadOutcome
 AwsDoc::S3::completeMultipartUpload(const Aws::String &bucket,

  const Aws::String &key,

  const Aws::String &uploadID,

  const Aws::Vector<Aws::S3::Model::CompletedPart> &parts,

  const Aws::S3::S3Client &client) {
    Aws::S3::Model::CompletedMultipartUpload completedMultipartUpload;
    completedMultipartUpload.SetParts(parts);

    Aws::S3::Model::CompleteMultipartUploadRequest request;
    request.SetBucket(bucket);
    request.SetKey(key);
    request.SetUploadId(uploadID);
    request.SetMultipartUpload(completedMultipartUpload);

    Aws::S3::Model::CompleteMultipartUploadOutcome outcome =
            client.CompleteMultipartUpload(request);

    if (!outcome.IsSuccess()) {
```

```
        std::cerr << "Error completing multipart upload: " <<
 outcome.GetError().GetMessage() << std::endl;
    }
    return outcome;
}

//! Routine which performs a multi-part upload.
/*!
    \param bucket: The name of the S3 bucket where the object will be uploaded.
    \param key: The unique identifier (key) for the object within the S3 bucket.
    \param hashMethod: The hashing algorithm to use when calculating the hash
 value.
    \param ioStream: An IOStream for the data to be uploaded.
    \param useDefaultHashMethod: A flag indicating whether to use the default
 hash method or the one specified in the hashMethod parameter.
    \param[out] hashDataResult: The Hasher object that will store the
 concatenated hash value.
    \param[out] partHashes: The vector that will store the calculated hash values
 for each part of the file.
    \param client: The S3 client instance used to perform the upload operation.
    \return bool: Function succeeded.
*/
bool AwsDoc::S3::doMultipartUpload(const Aws::String &bucket,
                                   const Aws::String &key,
                                   AwsDoc::S3::HASH_METHOD hashMethod,
                                   const std::shared_ptr<Aws::IOStream>
 &ioStream,
                                   bool useDefaultHashMethod,
                                   AwsDoc::S3::Hasher &hashDataResult,
                                   std::vector<Aws::String> &partHashes,
                                   const Aws::S3::S3Client &client) {
    // Get object size.
    ioStream->seekg(0, ioStream->end);
    size_t objectSize = ioStream->tellg();
    ioStream->seekg(0, ioStream->beg);

    Aws::S3::Model::ChecksumAlgorithm checksumAlgorithm =
 Aws::S3::Model::ChecksumAlgorithm::NOT_SET;
    if (!useDefaultHashMethod) {
        if (hashMethod != MD5) {
            checksumAlgorithm = getChecksumAlgorithmForHashMethod(hashMethod);
        }
    }
```

```
   Aws::String uploadID = createMultipartUpload(bucket, key, checksumAlgorithm,
client);
   if (uploadID.empty()) {
       return false;
   }

   std::vector<unsigned char> totalHashBuffer;
   bool uploadSucceeded = true;
   std::streamsize uploadedBytes = 0;
   int partNumber = 1;
   Aws::Vector<Aws::S3::Model::CompletedPart> parts;
   while (uploadedBytes < objectSize) {
       std::cout << "Uploading part " << partNumber << "." << std::endl;

       std::vector<unsigned char> buffer(UPLOAD_BUFFER_SIZE);
       std::streamsize bytesToRead =
static_cast<std::streamsize>(std::min(buffer.size(),

objectSize - uploadedBytes));
       ioStream->read((char *) buffer.data(), bytesToRead);
       Aws::Utils::Stream::PreallocatedStreamBuf
preallocatedStreamBuf(buffer.data(),

bytesToRead);
       std::shared_ptr<Aws::IOStream> body =
               Aws::MakeShared<Aws::IOStream>("SampleAllocationTag",
                                              &preallocatedStreamBuf);

       Hasher hasher;
       if (!hasher.calculateObjectHash(*body, hashMethod)) {
           std::cerr << "Error calculating hash." << std::endl;
           uploadSucceeded = false;
           break;
       }

       Aws::String base64HashString = hasher.getBase64HashString();
       partHashes.push_back(base64HashString);

       Aws::Utils::ByteBuffer hashBuffer = hasher.getByteBufferHash();

       totalHashBuffer.insert(totalHashBuffer.end(),
hashBuffer.GetUnderlyingData(),
                              hashBuffer.GetUnderlyingData() +
hashBuffer.GetLength());
```

```
        Aws::String calculatedHash;
        if (gUseCalculatedChecksum) {
            calculatedHash = base64HashString;
        }
        Aws::S3::Model::UploadPartOutcome uploadPartOutcome = uploadPart(bucket,
key, uploadID, partNumber,

checksumAlgorithm, base64HashString, body,
                                                                        client);
        if (uploadPartOutcome.IsSuccess()) {
            const Aws::S3::Model::UploadPartResult &uploadPartResult =
uploadPartOutcome.GetResult();
            Aws::S3::Model::CompletedPart completedPart;
            completedPart.SetETag(uploadPartResult.GetETag());
            completedPart.SetPartNumber(partNumber);
            switch (hashMethod) {
                case AwsDoc::S3::MD5:
                    break; // Do nothing.
                case AwsDoc::S3::SHA1:

completedPart.SetChecksumSHA1(uploadPartResult.GetChecksumSHA1());
                    break;
                case AwsDoc::S3::SHA256:

completedPart.SetChecksumSHA256(uploadPartResult.GetChecksumSHA256());
                    break;
                case AwsDoc::S3::CRC32:

completedPart.SetChecksumCRC32(uploadPartResult.GetChecksumCRC32());
                    break;
                case AwsDoc::S3::CRC32C:

completedPart.SetChecksumCRC32C(uploadPartResult.GetChecksumCRC32C());
                    break;
                default:
                    std::cerr << "Unhandled hash method for completedPart." <<
std::endl;
                    break;
            }

            parts.push_back(completedPart);
        } else {
            std::cerr << "Error uploading part. " <<
```

```
                        uploadPartOutcome.GetError().GetMessage() << std::endl;
            uploadSucceeded = false;
            break;
        }

        uploadedBytes += bytesToRead;
        partNumber++;
    }

    if (!uploadSucceeded) {
        abortMultipartUpload(bucket, key, uploadID, client);
        return false;
    } else {

        Aws::S3::Model::CompleteMultipartUploadOutcome
 completeMultipartUploadOutcome = completeMultipartUpload(bucket,

                                key,

                                uploadID,

                                parts,

                                client);

        if (completeMultipartUploadOutcome.IsSuccess()) {
            std::cout << "Multipart upload completed." << std::endl;
            if (!hashDataResult.calculateObjectHash(totalHashBuffer, hashMethod))
 {
                std::cerr << "Error calculating hash." << std::endl;
                return false;
            }
        } else {
            std::cerr << "Error completing multipart upload." <<
                      completeMultipartUploadOutcome.GetError().GetMessage()
                      << std::endl;
        }

        return completeMultipartUploadOutcome.IsSuccess();
    }
}

//! Routine which retrieves the string for a HASH_METHOD constant.
/*!
```

```
    \param: hashMethod: A HASH_METHOD constant.
    \return: String: A string description of the hash method.
*/
Aws::String AwsDoc::S3::stringForHashMethod(AwsDoc::S3::HASH_METHOD hashMethod) {
    switch (hashMethod) {
        case AwsDoc::S3::DEFAULT:
            return "Default";
        case AwsDoc::S3::MD5:
            return "MD5";
        case AwsDoc::S3::SHA1:
            return "SHA1";
        case AwsDoc::S3::SHA256:
            return "SHA256";
        case AwsDoc::S3::CRC32:
            return "CRC32";
        case AwsDoc::S3::CRC32C:
            return "CRC32C";
        default:
            return "Unknown";
    }
}

//! Routine that returns the ChecksumAlgorithm for a HASH_METHOD constant.
/*!
    \param: hashMethod: A HASH_METHOD constant.
    \return: ChecksumAlgorithm: The ChecksumAlgorithm enum.
*/
Aws::S3::Model::ChecksumAlgorithm
AwsDoc::S3::getChecksumAlgorithmForHashMethod(AwsDoc::S3::HASH_METHOD hashMethod)
 {
    Aws::S3::Model::ChecksumAlgorithm result =
 Aws::S3::Model::ChecksumAlgorithm::NOT_SET;
    switch (hashMethod) {
        case AwsDoc::S3::DEFAULT:
            std::cerr << "getChecksumAlgorithmForHashMethod- DEFAULT is not
 valid." << std::endl;
            break;  // Default is not supported.
        case AwsDoc::S3::MD5:
            break; // Ignore MD5.
        case AwsDoc::S3::SHA1:
            result = Aws::S3::Model::ChecksumAlgorithm::SHA1;
            break;
        case AwsDoc::S3::SHA256:
            result = Aws::S3::Model::ChecksumAlgorithm::SHA256;
```

```
            break;
        case AwsDoc::S3::CRC32:
            result = Aws::S3::Model::ChecksumAlgorithm::CRC32;
            break;
        case AwsDoc::S3::CRC32C:
            result = Aws::S3::Model::ChecksumAlgorithm::CRC32C;
            break;
        default:
            std::cerr << "Unknown hash method." << std::endl;
            break;

    }

    return result;
}

//! Routine which cleans up after the example is complete.
/*!
    \param bucket: The name of the S3 bucket where the object was uploaded.
    \param clientConfiguration: The client configuration for the S3 client.
    \return bool: Function succeeded.
*/
bool AwsDoc::S3::cleanUp(const Aws::String &bucketName,
                         const Aws::S3::S3ClientConfiguration
 &clientConfiguration) {

    Aws::Vector<Aws::String> keysResult;
    bool result = true;
    if (AwsDoc::S3::listObjects(bucketName, keysResult, clientConfiguration)) {
        if (!keysResult.empty()) {
            result = AwsDoc::S3::deleteObjects(keysResult, bucketName,
                                               clientConfiguration);
        }
    } else {
        result = false;
    }

    return result && AwsDoc::S3::deleteBucket(bucketName, clientConfiguration);
}

//! Console interaction introducing the workflow.
/*!
  \param bucketName: The name of the S3 bucket to use.
*/
```

```
void AwsDoc::S3::introductoryExplanations(const Aws::String &bucketName) {

   std::cout
           << "Welcome to the Amazon Simple Storage Service (Amazon S3) object
 integrity workflow."
           << std::endl;
   printAsterisksLine();
   std::cout
           << "This workflow demonstrates how Amazon S3 uses checksum values to
 verify the integrity of data\n";
   std::cout << "uploaded to Amazon S3 buckets" << std::endl;
   std::cout
           << "The AWS SDK for C++ automatically handles checksums.\n";
   std::cout
           << "By default it calculates a checksum that is uploaded with an
 object.\n"
           << "The default checksum algorithm for PutObject and MultiPart upload
 is an MD5 hash.\n"
           << "The default checksum algorithm for TransferManager uploads is a
 CRC32 checksum."
           << std::endl;
   std::cout
           << "You can override the default behavior, requiring one of the
 following checksums,\n";
   std::cout << "MD5, CRC32, CRC32C, SHA-1 or SHA-256." << std::endl;
   std::cout << "You can also set the checksum hash value, instead of letting
 the SDK calculate the value."
             << std::endl;
   std::cout
           << "For more information, see https://docs.aws.amazon.com/AmazonS3/
latest/userguide/checking-object-integrity.html."
           << std::endl;

   std::cout
           << "This workflow will locally compute checksums for files uploaded
 to an Amazon S3 bucket,\n";
   std::cout << "even when the SDK also computes the checksum." << std::endl;
   std::cout
           << "This is done to provide demonstration code for how the checksums
 are calculated."
           << std::endl;
   std::cout << "A bucket named '" << bucketName << "' will be created for the
 object uploads."
             << std::endl;
```

```
}

//! Console interaction which explains the PutObject results.
/*!
*/
void AwsDoc::S3::explainPutObjectResults() {

    std::cout << "The upload was successful.\n";
    std::cout << "If the checksums had not matched, the upload would have
 failed."
              << std::endl;
    std::cout
            << "The checksums calculated by the server have been retrieved using
 the GetObjectAttributes."
            << std::endl;
    std::cout
            << "The locally calculated checksums have been verified against the
 retrieved checksums."
            << std::endl;
}

//! Console interaction explaining transfer manager uploads.
/*!
  \param objectKey: The key for the object being uploaded.
*/
void AwsDoc::S3::introductoryTransferManagerUploadExplanations(
        const Aws::String &objectKey) {
    std::cout
            << "Now the workflow will demonstrate object integrity for
 TransferManager multi-part uploads."
            << std::endl;
    std::cout
            << "The AWS C++ SDK has a TransferManager class which simplifies
 multipart uploads."
            << std::endl;
    std::cout
            << "The following code lets the TransferManager handle much of the
 checksum configuration."
            << std::endl;

    std::cout << "An object with the key '" << objectKey
              << " will be uploaded by the TransferManager using a "
              << BUFFER_SIZE_IN_MEGABYTES << " MB buffer." << std::endl;
    if (gUseCalculatedChecksum) {
```

```
        std::cout << "For TransferManager uploads, this demo always lets the SDK
 calculate the hash value."
                  << std::endl;
    }

    pressEnterToContinue();
    printAsterisksLine();
}

//! Console interaction explaining multi-part uploads.
/*!
  \param objectKey: The key for the object being uploaded.
  \param chosenHashMethod: The hash method selected by the user.
*/
void AwsDoc::S3::multiPartUploadExplanations(const Aws::String &objectKey,
                                             HASH_METHOD chosenHashMethod) {
    std::cout
            << "Now we will provide an in-depth demonstration of multi-part
 uploading by calling the multi-part upload APIs directly."
            << std::endl;
    std::cout << "These are the same APIs used by the TransferManager when
 uploading large files."
              << std::endl;
    std::cout
            << "In the following code, the checksums are also calculated locally
 and then compared."
            << std::endl;
    std::cout
            << "For multi-part uploads, a checksum is uploaded with each part.
 The final checksum is a concatenation of"
            << std::endl;
    std::cout << "the checksums for each part." << std::endl;
    std::cout
            << "This is explained in the user guide, https://docs.aws.amazon.com/
AmazonS3/latest/userguide/checking-object-integrity.html,\""
            << " in the section \"Using part-level checksums for multipart
 uploads\"." << std::endl;

    std::cout << "Starting multipart upload of with hash method " <<
              stringForHashMethod(chosenHashMethod) << " uploading to with object
 key\n"
              << "'" << objectKey << "'," << std::endl;

}
```

```
//! Create a large file for doing multi-part uploads.
/*!
*/
bool AwsDoc::S3::createLargeFileIfNotExists() {
    // Generate a large file by writing this source file multiple times to a new
 file.
    if (std::filesystem::exists(MULTI_PART_TEST_FILE)) {
        return true;
    }

    std::ofstream newFile(MULTI_PART_TEST_FILE, std::ios::out

                                                | std::ios::binary);

    if (!newFile) {
        std::cerr << "createLargeFileIfNotExists- Error creating file " <<
 MULTI_PART_TEST_FILE <<
                  std::endl;
        return false;
    }

    std::ifstream input(TEST_FILE, std::ios::in

                                   | std::ios::binary);
    if (!input) {
        std::cerr << "Error opening file " << TEST_FILE <<
                  std::endl;
        return false;
    }
    std::stringstream buffer;
    buffer << input.rdbuf();

    input.close();

    while (newFile.tellp() < LARGE_FILE_SIZE && !newFile.bad()) {
        buffer.seekg(std::stringstream::beg);
        newFile << buffer.rdbuf();
    }

    newFile.close();

    return true;
}
```

- Para obtener información sobre la API, consulte los siguientes temas en la referencia de la API de AWS SDK for C++.
  - AbortMultipartUpload
  - CompleteMultipartUpload
  - CreateMultipartUpload
  - DeleteObject
  - GetObjectAttributes
  - PutObject
  - UploadPart

Para obtener una lista completa de las guías para desarrolladores del AWS SDK y ejemplos de código, consulte Uso de este servicio con un SDK de AWS. En este tema también se incluye información sobre cómo comenzar a utilizar el SDK y detalles sobre sus versiones anteriores.

Trabajo con objetos con control de versiones de Amazon S3 con un SDK de AWS

En el siguiente ejemplo de código, se muestra cómo:

- Crear un bucket de S3 con control de versiones.
- Obtener todas las versiones de un objeto.
- Revertir un objeto a una versión anterior.
- Eliminar y restaurar un objeto con control de versiones.
- Eliminar permanentemente todas las versiones de un objeto.

Python

SDK para Python (Boto3)

> **Note**
>
> Hay más en GitHub. Busque el ejemplo completo y aprenda a configurar y ejecutar en el Repositorio de ejemplos de código de AWS.

Crear funciones que encapsulen acciones de 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
```

Cargar la estrofa de un poema en un objeto con control de versiones y realizar una serie de acciones en él.

```
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!")
```

- Para obtener información sobre la API, consulte los siguientes temas en la Referencia de la API de AWS SDK para Python (Boto3).
  - CreateBucket
  - DeleteObject
  - ListObjectVersions
  - PutBucketLifecycleConfiguration

Para obtener una lista completa de las guías para desarrolladores del AWS SDK y ejemplos de código, consulte Uso de este servicio con un SDK de AWS. En este tema también se incluye información sobre cómo comenzar a utilizar el SDK y detalles sobre sus versiones anteriores.

Ejemplos sin servidor para Amazon S3 que utilizan SDK de AWS

Los siguientes ejemplos de código muestran cómo utilizar Amazon S3 con los SDK de AWS.

Ejemplos

- Invocación de una función de Lambda desde un desencadenador de Amazon S3

Invocación de una función de Lambda desde un desencadenador de Amazon S3

En los siguientes ejemplos de código, se muestra cómo implementar una función de Lambda que recibe un evento desencadenado al cargar un objeto en un bucket de S3. La función recupera el nombre del bucket de S3 y la clave del objeto del parámetro de evento y llama a la API de Amazon S3 para recuperar y registrar el tipo de contenido del objeto.

.NET

AWS SDK for .NET

> **Note**
>
> Hay más información en GitHub. Busque el ejemplo completo y aprenda a configurar y ejecutar en el repositorio de ejemplos sin servidor.

Uso de un evento de S3 con Lambda mediante .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

SDK para Go V2

> **Note**
>
> Hay más información en GitHub. Busque el ejemplo completo y aprenda a configurar y ejecutar en el repositorio de ejemplos sin servidor.

Uso de un evento de S3 con Lambda mediante 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 para Java 2.x

> **Note**
>
> Hay más información en GitHub. Busque el ejemplo completo y aprenda a configurar y ejecutar en el repositorio de ejemplos sin servidor.

Uso de un evento de S3 con Lambda mediante Java.

```
// 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 para JavaScript (v3)

> **Note**
>
> Hay más información en GitHub. Busque el ejemplo completo y aprenda a configurar y ejecutar en el repositorio de ejemplos sin servidor.

Uso de un evento de S3 con Lambda mediante 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);
    }
};
```

Uso de un evento de S3 con Lambda mediante 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

SDK para PHP

> **Note**
>
> Hay más información en GitHub. Busque el ejemplo completo y aprenda a configurar y ejecutar en el repositorio de ejemplos sin servidor.

Consumo de un evento de S3 con Lambda mediante 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 para Python (Boto3)

> **Note**
>
> Hay más información en GitHub. Busque el ejemplo completo y aprenda a configurar y ejecutar en el repositorio de ejemplos sin servidor.

Uso de un evento de S3 con Lambda mediante 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

SDK para Ruby

> **Note**
>
> Hay más información en GitHub. Busque el ejemplo completo y aprenda a configurar y ejecutar en el repositorio de ejemplos sin servidor.

Consumo de un evento de S3 con Lambda mediante 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 para Rust

> **Note**
>
> Hay más información en GitHub. Busque el ejemplo completo y aprenda a configurar y ejecutar en el repositorio de ejemplos sin servidor.

Uso de un evento de S3 con Lambda mediante 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(())
}
```

Para obtener una lista completa de las guías para desarrolladores del AWS SDK y ejemplos de código, consulte Uso de este servicio con un SDK de AWS. En este tema también se incluye información sobre cómo comenzar a utilizar el SDK y detalles sobre sus versiones anteriores.

Ejemplos de servicios combinados de Amazon S3 con SDK de AWS

Las siguientes aplicaciones de ejemplo utilizan SDK de AWS para combinar Amazon S3 con otros Servicios de AWS. Cada ejemplo incluye un enlace a GitHub, con instrucciones de configuración y ejecución de la aplicación.

Ejemplos

- Cree una aplicación Amazon Transcribe
- Convierta texto en voz y de nuevo a texto con un SDK de AWS
- Creación de una aplicación de administración de activos fotográficos que permita a los usuarios administrar las fotos mediante etiquetas
- Creación de una aplicación de exploración de Amazon Textract
- Detección de EPI en imágenes con Amazon Rekognition mediante un AWS SDK
- Detecte entidades en el texto extraído de una imagen con un SDK de AWS
- Detecte rostros en una imagen con un SDK de AWS
- Detección de personas y objetos en un vídeo con Amazon Rekognition mediante un AWS SDK
- Detecte personas y objetos en un vídeo con Amazon Rekognition mediante un SDK de AWS
- Guarde EXIF y otra información de la imagen con un SDK de AWS
- Transformación de datos para su aplicación con S3 Object Lambda

Cree una aplicación Amazon Transcribe

En el siguiente ejemplo de código, se muestra cómo utilizar Amazon Transcribe para transcribir y mostrar grabaciones de voz en el navegador.

JavaScript

SDK para JavaScript (v3)

Cree una aplicación que utilice Amazon Transcribe para transcribir y mostrar grabaciones de voz en el navegador. La aplicación utiliza dos buckets de Amazon Simple Storage Service (Amazon S3), uno para alojar el código de la aplicación y otro para almacenar transcripciones. La aplicación utiliza un grupo de usuarios de Amazon Cognito para autenticar a los usuarios.

Los usuarios autenticados tienen permisos de AWS Identity and Access Management (IAM) para obtener acceso a los servicios de AWS requeridos.

Para ver el código fuente completo y las instrucciones de configuración y ejecución, consulte el ejemplo completo en GitHub.

Este ejemplo también está disponible en la guía para desarrolladores de AWS SDK for JavaScript v3.

Servicios utilizados en este ejemplo

- Amazon Cognito Identity
- Amazon S3
- Amazon Transcribe

Para obtener una lista completa de las guías para desarrolladores del AWS SDK y ejemplos de código, consulte Uso de este servicio con un SDK de AWS. En este tema también se incluye información sobre cómo comenzar a utilizar el SDK y detalles sobre sus versiones anteriores.

Convierta texto en voz y de nuevo a texto con un SDK de AWS

En el siguiente ejemplo de código, se muestra cómo:

- Utilice Amazon Polly para sintetizar un archivo de entrada de texto sin formato (UTF-8) en un archivo de audio.
- Cargue el archivo de audio en un bucket de Amazon S3.
- Utilice Amazon Transcribe para convertir el archivo de audio en texto.
- Muestre el texto.

Rust

SDK para Rust

Utilice Amazon Polly para sintetizar un archivo de entrada de texto sin formato (UTF-8) en un archivo de audio, cargue el archivo de audio en un bucket de Amazon S3, utilice Amazon Transcribe para convertir ese archivo de audio en texto y muestre el texto.

Para ver el código fuente completo y las instrucciones de configuración y ejecución, consulte el ejemplo completo en GitHub.

Servicios utilizados en este ejemplo

- Amazon Polly
- Amazon S3
- Amazon Transcribe

Para obtener una lista completa de las guías para desarrolladores del AWS SDK y ejemplos de código, consulte Uso de este servicio con un SDK de AWS. En este tema también se incluye información sobre cómo comenzar a utilizar el SDK y detalles sobre sus versiones anteriores.

Creación de una aplicación de administración de activos fotográficos que permita a los usuarios administrar las fotos mediante etiquetas

En los siguientes ejemplos de código se muestra cómo crear una aplicación sin servidor que permita a los usuarios administrar fotos mediante etiquetas.

.NET

AWS SDK for .NET

Muestra cómo desarrollar una aplicación de gestión de activos fotográficos que detecte las etiquetas de las imágenes mediante Amazon Rekognition y las almacene para su posterior recuperación.

Para ver el código fuente completo y las instrucciones de configuración y ejecución, consulte el ejemplo completo en GitHub.

Para profundizar en el origen de este ejemplo, consulte la publicación en Comunidad de AWS.

Servicios utilizados en este ejemplo

- API Gateway
- DynamoDB
- Lambda
- Amazon Rekognition
- Amazon S3
- Amazon SNS

C++

SDK para C++

Muestra cómo desarrollar una aplicación de gestión de activos fotográficos que detecte las etiquetas de las imágenes mediante Amazon Rekognition y las almacene para su posterior recuperación.

Para ver el código fuente completo y las instrucciones de configuración y ejecución, consulte el ejemplo completo en GitHub.

Para profundizar en el origen de este ejemplo, consulte la publicación en Comunidad de AWS.

Servicios utilizados en este ejemplo

- API Gateway
- DynamoDB
- Lambda
- Amazon Rekognition
- Amazon S3
- Amazon SNS

Java

SDK para Java 2.x

Muestra cómo desarrollar una aplicación de gestión de activos fotográficos que detecte las etiquetas de las imágenes mediante Amazon Rekognition y las almacene para su posterior recuperación.

Para ver el código fuente completo y las instrucciones de configuración y ejecución, consulte el ejemplo completo en GitHub.

Para profundizar en el origen de este ejemplo, consulte la publicación en Comunidad de AWS.

Servicios utilizados en este ejemplo

- API Gateway
- DynamoDB
- Lambda

- Amazon Rekognition
- Amazon S3
- Amazon SNS

JavaScript

SDK para JavaScript (v3)

Muestra cómo desarrollar una aplicación de gestión de activos fotográficos que detecte las etiquetas de las imágenes mediante Amazon Rekognition y las almacene para su posterior recuperación.

Para ver el código fuente completo y las instrucciones de configuración y ejecución, consulte el ejemplo completo en GitHub.

Para profundizar en el origen de este ejemplo, consulte la publicación en Comunidad de AWS.

Servicios utilizados en este ejemplo

- API Gateway
- DynamoDB
- Lambda
- Amazon Rekognition
- Amazon S3
- Amazon SNS

Kotlin

SDK para Kotlin

Muestra cómo desarrollar una aplicación de gestión de activos fotográficos que detecte las etiquetas de las imágenes mediante Amazon Rekognition y las almacene para su posterior recuperación.

Para ver el código fuente completo y las instrucciones de configuración y ejecución, consulte el ejemplo completo en GitHub.

Para profundizar en el origen de este ejemplo, consulte la publicación en Comunidad de AWS.

Servicios utilizados en este ejemplo

- API Gateway
- DynamoDB
- Lambda
- Amazon Rekognition
- Amazon S3
- Amazon SNS

PHP

SDK para PHP

Muestra cómo desarrollar una aplicación de gestión de activos fotográficos que detecte las etiquetas de las imágenes mediante Amazon Rekognition y las almacene para su posterior recuperación.

Para ver el código fuente completo y las instrucciones de configuración y ejecución, consulte el ejemplo completo en GitHub.

Para profundizar en el origen de este ejemplo, consulte la publicación en Comunidad de AWS.

Servicios utilizados en este ejemplo

- API Gateway
- DynamoDB
- Lambda
- Amazon Rekognition
- Amazon S3
- Amazon SNS

Rust

SDK para Rust

Muestra cómo desarrollar una aplicación de gestión de activos fotográficos que detecte las etiquetas de las imágenes mediante Amazon Rekognition y las almacene para su posterior recuperación.

Para ver el código fuente completo y las instrucciones de configuración y ejecución, consulte el ejemplo completo en GitHub.

Para profundizar en el origen de este ejemplo, consulte la publicación en Comunidad de AWS.

Servicios utilizados en este ejemplo

- API Gateway
- DynamoDB
- Lambda
- Amazon Rekognition
- Amazon S3
- Amazon SNS

Para obtener una lista completa de las guías para desarrolladores del SDK de AWS y ejemplos de código, consulte Uso de este servicio con un SDK de AWS. En este tema también se incluye información sobre cómo comenzar a utilizar el SDK y detalles sobre sus versiones anteriores.

Creación de una aplicación de exploración de Amazon Textract

Los siguientes ejemplos de código indican cómo explorar la salida de Amazon Textract mediante una aplicación interactiva.

JavaScript

SDK para JavaScript (v3)

Indica cómo utilizar el AWS SDK for JavaScript para crear una aplicación React que utilice Amazon Textract para extraer datos de la imagen de un documento y presentarlos en una página web interactiva. Este ejemplo se ejecuta en un navegador web y requiere una identidad autenticada de Amazon Cognito para las credenciales. Para el almacenamiento utiliza Amazon Simple Storage Service (Amazon S3) y para las notificaciones consulta una cola de Amazon Simple Queue Service (Amazon SQS) que está suscrita a un tema de Amazon Simple Notification Service (Amazon SNS).

Para ver el código fuente completo y las instrucciones de configuración y ejecución, consulte el ejemplo completo en GitHub.

Servicios utilizados en este ejemplo

- Amazon Cognito Identity
- Amazon S3
- Amazon SNS
- Amazon SQS
- Amazon Textract

Python

SDK para Python (Boto3)

Indica cómo utilizar AWS SDK for Python (Boto3) con Amazon Textract para detectar elementos de texto, formularios y tablas en la imagen de un documento. La imagen de entrada y la salida de Amazon Textract aparecen en una aplicación Tkinter que permite explorar los elementos detectados.

- Envía la imagen de un documento a Amazon Textract y explora el resultado de los elementos detectados.
- Envía imágenes directamente a Amazon Textract o mediante un bucket de Amazon Simple Storage Service (Amazon S3).
- Utilice las API asíncronas para iniciar un trabajo que publique una notificación en un tema de Amazon Simple Notification Service (Amazon SNS) cuando el trabajo se finalice.
- Consulta una cola de Amazon Simple Queue Service (Amazon SQS) en busca de un mensaje de finalización de trabajo y muestra los resultados.

Para ver el código fuente completo y las instrucciones de configuración y ejecución, consulte el ejemplo completo en GitHub.

Servicios utilizados en este ejemplo

- Amazon S3
- Amazon SNS
- Amazon SQS
- Amazon Textract

Para obtener una lista completa de las guías para desarrolladores del SDK de AWS y ejemplos de código, consulte Uso de este servicio con un SDK de AWS. En este tema también se incluye información sobre cómo comenzar a utilizar el SDK y detalles sobre sus versiones anteriores.

Detección de EPI en imágenes con Amazon Rekognition mediante un AWS SDK

Los siguientes ejemplos de código muestran cómo crear una aplicación que utiliza Amazon Rekognition para detectar equipos de protección individual (EPI) en imágenes.

Java

SDK para Java 2.x

Muestra cómo crear una función de AWS Lambda que detecte imágenes con equipos de protección individual.

Para ver el código fuente completo y las instrucciones de configuración y ejecución, consulte el ejemplo completo en GitHub.

Servicios utilizados en este ejemplo

- DynamoDB
- Amazon Rekognition
- Amazon S3
- Amazon SES

JavaScript

SDK para JavaScript (v3)

Muestra cómo utilizar Amazon Rekognition con AWS SDK for JavaScript para crear una aplicación que detecte equipos de protección individual (EPI) en imágenes ubicadas en un bucket de Amazon Simple Storage Service (Amazon S3). La aplicación guarda los resultados en una tabla de Amazon DynamoDB y envía al administrador una notificación por correo electrónico con los resultados mediante Amazon Simple Email Service (Amazon SES).

Aprenda cómo:

- Crear un usuario no autenticado con Amazon Cognito.

- Analizar imágenes en busca de EPI con Amazon Rekognition.
- Verificar una dirección de correo electrónico de Amazon SES.
- Actualizar una tabla de DynamoDB con resultados.
- Enviar una notificación por correo electrónico con Amazon SES.

Para ver el código fuente completo y las instrucciones de configuración y ejecución, consulte el ejemplo completo en GitHub.

Servicios utilizados en este ejemplo

- DynamoDB
- Amazon Rekognition
- Amazon S3
- Amazon SES

Para obtener una lista completa de las guías para desarrolladores del AWS SDK y ejemplos de código, consulte Uso de este servicio con un SDK de AWS. En este tema también se incluye información sobre cómo comenzar a utilizar el SDK y detalles sobre sus versiones anteriores.

Detecte entidades en el texto extraído de una imagen con un SDK de AWS

En el siguiente ejemplo de código, se muestra cómo utilizar Amazon Comprehend para detectar entidades en el texto extraído por Amazon Textract Texact de una imagen almacenada en Amazon S3.

Python

SDK para Python (Boto3)

Muestra cómo utilizar AWS SDK for Python (Boto3) en un bloc de notas de Jupyter para detectar entidades del texto que se extrae de una imagen. En este ejemplo, se utiliza Amazon Textract para extraer texto de una imagen almacenada en Amazon Simple Storage Service (Amazon S3) y Amazon Comprehend para detectar entidades en el texto extraído.

Este ejemplo es un bloc de notas Jupyter y debe ejecutarse en un entorno que pueda alojar blocs de notas. Para obtener instrucciones sobre cómo ejecutar el ejemplo con Amazon SageMaker, consulte las instrucciones en TextractAndComprehendNotebook.ipynb.

Para ver el código fuente completo y las instrucciones de configuración y ejecución, consulte el ejemplo completo en GitHub.

Servicios utilizados en este ejemplo

- Amazon Comprehend
- Amazon S3
- Amazon Textract

Para obtener una lista completa de las guías para desarrolladores del SDK de AWS y ejemplos de código, consulte Uso de este servicio con un SDK de AWS. En este tema también se incluye información sobre cómo comenzar a utilizar el SDK y detalles sobre sus versiones anteriores.

Detecte rostros en una imagen con un SDK de AWS

En el siguiente ejemplo de código, se muestra cómo:

- Guarde una imagen en un bucket de Amazon S3.
- Utilice Amazon Rekognition para detectar información faciales, como el rango de edad, el género y las emociones (por ejemplo, una sonrisa).
- Muestre esos detalles.

Rust

SDK para Rust

Guarde la imagen en un bucket de Amazon S3 con el prefijo uploads, use Amazon Rekognition para detectar información faciales, como el rango de edad, el género y las emociones (por ejemplo, una sonrisa) y muestre esos detalles.

Para ver el código fuente completo y las instrucciones de configuración y ejecución, consulte el ejemplo completo en GitHub.

Servicios utilizados en este ejemplo

- Amazon Rekognition
- Amazon S3

Para obtener una lista completa de las guías para desarrolladores del AWS SDK y ejemplos de código, consulte Uso de este servicio con un SDK de AWS. En este tema también se incluye información sobre cómo comenzar a utilizar el SDK y detalles sobre sus versiones anteriores.

Detección de personas y objetos en un vídeo con Amazon Rekognition mediante un AWS SDK

Los siguientes ejemplos de código muestran cómo crear una aplicación que utilice Amazon Rekognition para detectar objetos por categoría en imágenes.

.NET

AWS SDK for .NET

Muestra cómo utilizar la API de .NET de Amazon Rekognition para crear una aplicación que utilice Amazon Rekognition para identificar objetos por categoría en imágenes ubicadas en un bucket de Amazon Simple Storage Service (Amazon S3). La aplicación envía al administrador una notificación por correo electrónico con los resultados mediante Amazon Simple Email Service (Amazon SES).

Para ver el código fuente completo y las instrucciones de configuración y ejecución, consulte el ejemplo completo en GitHub.

Servicios utilizados en este ejemplo

- Amazon Rekognition
- Amazon S3
- Amazon SES

Java

SDK para Java 2.x

Muestra cómo utilizar la API de Java de Amazon Rekognition para crear una aplicación que utilice Amazon Rekognition para identificar objetos por categoría en imágenes ubicadas en un bucket de Amazon Simple Storage Service (Amazon S3). La aplicación envía al administrador una notificación por correo electrónico con los resultados mediante Amazon Simple Email Service (Amazon SES).

Para ver el código fuente completo y las instrucciones de configuración y ejecución, consulte el ejemplo completo en GitHub.

Servicios utilizados en este ejemplo

- Amazon Rekognition
- Amazon S3
- Amazon SES

JavaScript

SDK para JavaScript (v3)

Muestra cómo utilizar Amazon Rekognition con AWS SDK for JavaScript para crear una aplicación que utilice Amazon Rekognition para identificar objetos por categoría en imágenes ubicadas en un bucket de Amazon Simple Storage Service (Amazon S3). La aplicación envía al administrador una notificación por correo electrónico con los resultados mediante Amazon Simple Email Service (Amazon SES).

Aprenda cómo:

- Crear un usuario no autenticado con Amazon Cognito.
- Analizar imágenes en busca de objetos con Amazon Rekognition.
- Verificar una dirección de correo electrónico de Amazon SES.
- Enviar una notificación por correo electrónico con Amazon SES.

Para ver el código fuente completo y las instrucciones de configuración y ejecución, consulte el ejemplo completo en GitHub.

Servicios utilizados en este ejemplo

- Amazon Rekognition
- Amazon S3
- Amazon SES

Kotlin

SDK para Kotlin

Muestra cómo utilizar la API de Kotlin de Amazon Rekognition para crear una aplicación que utilice Amazon Rekognition para identificar objetos por categoría en imágenes ubicadas en un bucket de Amazon Simple Storage Service (Amazon S3). La aplicación envía al administrador una notificación por correo electrónico con los resultados mediante Amazon Simple Email Service (Amazon SES).

Para ver el código fuente completo y las instrucciones de configuración y ejecución, consulte el ejemplo completo en GitHub.

Servicios utilizados en este ejemplo

- Amazon Rekognition
- Amazon S3
- Amazon SES

Python

SDK para Python (Boto3)

Le muestra cómo utilizar AWS SDK for Python (Boto3) para crear una aplicación web que le permita hacer lo siguiente:

- Subir fotos en un bucket de Amazon Simple Storage Service (Amazon S3).
- Utilizar Amazon Rekognition para analizar y etiquetar las fotos.
- Utilice Amazon Simple Email Service (Amazon SES) para enviar informes de análisis de imágenes por correo electrónico.

Este ejemplo contiene dos componentes principales: una página web escrita en JavaScript creada con React y un servicio REST escrito en Python creado con Flask-RESTful.

Puede utilizar la página web de React para:

- Mostrar una lista de imágenes almacenadas en el bucket de S3.
- Subir imágenes desde la computadora en el bucket de S3.
- Mostrar imágenes y etiquetas que identifican los elementos detectados en la imagen.

- Obtener un informe de todas las imágenes del bucket de S3 y enviar un correo electrónico del informe.

La página web llama al servicio REST. El servicio envía solicitudes a AWS para llevar a cabo las siguientes acciones:

- Obtener y filtrar la lista de imágenes del bucket de S3.
- Subir fotos en el bucket de S3.
- Utilizar Amazon Rekognition para analizar fotos individuales y obtener una lista de etiquetas que identifican los elementos detectados en la foto.
- Analizar todas las fotos del bucket de S3 y usar Amazon SES para enviar un informe por correo electrónico.

Para ver el código fuente completo y las instrucciones de configuración y ejecución, consulte el ejemplo completo en GitHub.

Servicios utilizados en este ejemplo

- Amazon Rekognition
- Amazon S3
- Amazon SES

Para obtener una lista completa de las guías para desarrolladores del AWS SDK y ejemplos de código, consulte Uso de este servicio con un SDK de AWS. En este tema también se incluye información sobre cómo comenzar a utilizar el SDK y detalles sobre sus versiones anteriores.

Detecte personas y objetos en un vídeo con Amazon Rekognition mediante un SDK de AWS

Los siguientes ejemplos de código indican cómo detectar personas y objetos en un video con Amazon Rekognition.

Java

SDK para Java 2.x

Muestra cómo utilizar la API Java de Amazon Rekognition para crear una aplicación que detecte rostros y objetos en vídeos ubicados en un bucket de Amazon Simple Storage Service

(Amazon S3). La aplicación envía al administrador una notificación por correo electrónico con los resultados mediante Amazon Simple Email Service (Amazon SES).

Para ver el código fuente completo y las instrucciones de configuración y ejecución, consulte el ejemplo completo en GitHub.

Servicios utilizados en este ejemplo

- Amazon Rekognition
- Amazon S3
- Amazon SES

JavaScript

SDK para JavaScript (v3)

Muestra cómo utilizar Amazon Rekognition con AWS SDK for JavaScript para crear una aplicación que detecte rostros y objetos en vídeos ubicados en un bucket de Amazon Simple Storage Service (Amazon S3). La aplicación envía al administrador una notificación por correo electrónico con los resultados mediante Amazon Simple Email Service (Amazon SES).

Aprenda cómo:

- Crear un usuario no autenticado con Amazon Cognito.
- Analizar imágenes en busca de EPI con Amazon Rekognition.
- Verificar una dirección de correo electrónico de Amazon SES.
- Enviar una notificación por correo electrónico con Amazon SES.

Para ver el código fuente completo y las instrucciones de configuración y ejecución, consulte el ejemplo completo en GitHub.

Servicios utilizados en este ejemplo

- Amazon Rekognition
- Amazon S3
- Amazon SES

Para obtener una lista completa de las guías para desarrolladores del AWS SDK y ejemplos de código, consulte Uso de este servicio con un SDK de AWS. En este tema también se incluye información sobre cómo comenzar a utilizar el SDK y detalles sobre sus versiones anteriores.

Guarde EXIF y otra información de la imagen con un SDK de AWS

En el siguiente ejemplo de código, se muestra cómo:

- Obtenga información EXIF de un archivo JPG, JPEG o PNG.
- Subir el archivo de imagen en un bucket de Amazon S3.
- Usar Amazon Rekognition para identificar los tres atributos principales (etiquetas) en el archivo.
- Agregar la información EXIF y de etiquetas a una tabla de Amazon DynamoDB de la región.

Rust

SDK para Rust

Obtenga información EXIF de un archivo JPG, JPEG o PNG, cargue el archivo de imagen en un bucket de Amazon S3, utilice Amazon Rekognition para identificar los tres atributos principales (etiquetas de Amazon Rekognition) en el archivo y añada la información EXIF y de etiquetas a una tabla de Amazon DynamoDB de la región.

Para ver el código fuente completo y las instrucciones de configuración y ejecución, consulte el ejemplo completo en GitHub.

Servicios utilizados en este ejemplo

- DynamoDB
- Amazon Rekognition
- Amazon S3

Para obtener una lista completa de las guías para desarrolladores del AWS SDK y ejemplos de código, consulte Uso de este servicio con un SDK de AWS. En este tema también se incluye información sobre cómo comenzar a utilizar el SDK y detalles sobre sus versiones anteriores.

Transformación de datos para su aplicación con S3 Object Lambda

En el siguiente ejemplo de código se muestra cómo transformar datos para su aplicación con S3 Object Lambda.

.NET

AWS SDK for .NET

Muestra cómo agregar código personalizado a las solicitudes GET S3 estándar para modificar el objeto solicitado recuperado de S3, de modo que el objeto se ajuste a las necesidades del cliente o aplicación solicitante.

Para ver el código fuente completo y las instrucciones de configuración y ejecución, consulte el ejemplo completo en GitHub.

Servicios utilizados en este ejemplo

- Lambda
- Amazon S3

Para obtener una lista completa de las guías para desarrolladores del AWS SDK y ejemplos de código, consulte Uso de este servicio con un SDK de AWS. En este tema también se incluye información sobre cómo comenzar a utilizar el SDK y detalles sobre sus versiones anteriores.

Solución de problemas

En esta sección, se describe cómo solucionar los problemas de Amazon S3 y se explica cómo obtener los ID de solicitudes que necesitará para contactar con AWS Support.

Temas

Solucionar errores de acceso denegado (403 Prohibido) en Amazon S3

> **Important**
>
> El 13 de mayo de 2024, empezamos a implementar un cambio para eliminar los cargos por solicitudes no autorizadas que no haya iniciado el propietario del bucket. Una vez que se complete la implementación de este cambio, los propietarios de los buckets nunca incurrirán en cargos por solicitud o ancho de banda por las solicitudes que devuelvan errores `AccessDenied` (HTTP `403 Forbidden`) cuando estas solicitudes se inicien desde fuera de la cuenta de AWS individual u organización de AWS. Para obtener más información sobre una lista completa de códigos de estado `3XX` y `4XX` HTTP que no se facturarán, consulte Facturación para respuestas de errores de Amazon S3. Este cambio de facturación no requiere actualizaciones en las aplicaciones y se aplica a todos los buckets de S3. Cuando se haya completado la implementación de este cambio en todas las Regiones de AWS, actualizaremos nuestra documentación.

En los siguientes temas se describen las causas más comunes de los errores de acceso denegado (403 Prohibido) en Amazon S3.

> **Note**
>
> Para `Access Denied` (HTTP `403 Forbidden`), S3 no cobra al propietario del bucket cuando la solicitud se inicia fuera de la cuenta de AWS individual del propietario del bucket o de la organización de AWS del propietario del bucket.

Temas

> **Note**
>
> Si está intentando solucionar un problema de permisos, empiece por la sección Bucket policies and IAM policies (Políticas de buckets y políticas de IAM) y asegúrese de seguir las indicaciones de Tips for checking permissions (Consejos para comprobar los permisos).

Políticas de bucket y de IAM

Operaciones de bucket

Si no existe una política de bucket, el bucket permite de forma implícita las solicitudes de cualquier identidad AWS Identity and Access Management (IAM) de la cuenta propietaria del bucket. El bucket también rechaza implícitamente las solicitudes de cualquier otra identidad de IAM de cualquier otra cuenta y las solicitudes anónimas (sin firma). Sin embargo, si no existe una política de usuario de IAM, se niega implícitamente al solicitante (a menos que sea el usuario raíz) realizar ninguna solicitud. Para obtener más información sobre esta lógica de evaluación, consulte Cómo determinar si una solicitud se permite o se deniega dentro de una cuenta en la Guía del usuario de IAM.

Operaciones en el nivel de ls objeto

Si el objeto es propiedad de la cuenta propietaria del bucket, la política de bucket y la política de usuario de IAM funcionarán de la misma manera para las operaciones en el nivel de objeto que para las operaciones en el nivel de bucket. Por ejemplo, si no existe una política de bucket, el bucket permite de forma implícita las solicitudes de objetos de cualquier identidad de IAM de la cuenta propietaria del bucket. El bucket también rechaza implícitamente las solicitudes de objeto de cualquier otra identidad de IAM de cualquier otra cuenta y las solicitudes anónimas (sin firma). Sin embargo, si no existe una política de usuario de IAM, se niega implícitamente al solicitante realizar solicitudes de objetos (a menos que sea el usuario raíz).

Si el objeto es propiedad de una cuenta externa, el acceso al objeto solo se puede conceder a través de listas de control de acceso (ACL) a objetos. La política de bucket y la política de usuario de IAM todavía se pueden utilizar para denegar las solicitudes de objetos.

Por lo tanto, para garantizar que su política de bucket o de usuarios de IAM no esté provocando un error de acceso denegado (403 Prohibido), debe cumplir los siguientes requisitos:

- Para acceder con la misma cuenta, no debe haber ninguna instrucción `Deny` explícita para el solicitante al que intenta conceder permisos, ni en la política de bucket ni en la política de usuarios de IAM. Si desea conceder permisos utilizando únicamente la política de bucket y la política de usuarios de IAM, debe haber al menos una instrucción `Allow` explícita en una de estas políticas.
- Para un acceso entre cuentas, no debe haber ninguna instrucción `Deny` explícita para el solicitante al que intenta conceder permisos, ni en la política de bucket ni en la política de usuarios de IAM. Si desea conceder permisos entre cuentas utilizando únicamente la política de bucket y la política de usuarios de IAM, tanto la política de bucket como la política de usuarios de IAM del solicitante deben incluir una instrucción `Allow` explícita.

> **Note**
>
> Las instrucciones `Allow` de una política de bucket se aplican solo a los objetos que pertenecen a la misma cuenta propietaria del bucket. Sin embargo, las instrucciones `Deny` de una política de bucket se aplican a todos los objetos independientemente de quién sea el propietario.

Para revisar o editar la política de bucket

> **Note**
>
> Para ver o editar una política de bucket, debe tener el permiso `s3:GetBucketPolicy`.

1. Inicie sesión AWS Management Console Management Console y abra la consola de Amazon S3 en https://console.aws.amazon.com/s3/.
2. En el panel de navegación izquierdo, elija Instancias.
3. En la lista Buckets, seleccione el nombre del bucket para el que desea crear una política de bucket.
4. Elija la pestaña Permisos.
5. En Política de bucket, elija Editar. Aparece la página Editar política de bucket.

Para revisar o editar la política de bucket mediante la AWS Command Line Interface (AWS CLI), use el comando get-bucket-policy.

> **Note**
>
> Si tiene bloqueado el acceso a un bucket debido a una política de bucket incorrecta, inicie sesión en la AWS Management Console con las credenciales de usuario raíz. Para recuperar el acceso al bucket, asegúrese de eliminar la política de bucket con sus credenciales de usuario raíz.

Consejos para comprobar los permisos

Para comprobar si el solicitante tiene los permisos adecuados para realizar una operación de Amazon S3, pruebe lo siguiente:

- Identifique al solicitante. Si se trata de una solicitud sin firma, significa que es una solicitud anónima sin política de usuarios de IAM. Si se trata de una solicitud que utiliza una URL prefirmada, la política de usuario será la misma que la del usuario o rol de IAM que firmó la solicitud.

- Verifique que está usando el usuario o el rol de IAM correctos. Para comprobar su usuario o rol de IAM, consulte la esquina superior derecha de la AWS Management Console o utilice el comando aws sts get-caller-identity.
- Compruebe las políticas de IAM asociadas al usuario o rol de IAM. Puede usar uno de los métodos siguientes:
  - Probar las políticas de IAM con el simulador de política de IAM.
  - Revisar los diferentes tipos de políticas de IAM.
- Si es necesario, edite la política de usuario de IAM.
- Revise los siguientes ejemplos de políticas que deniegan o permiten el acceso de forma explícita:
  - Permitir la política de usuarios de IAM de forma explícita: IAM: permite y deniega el acceso a varios servicios mediante programación y en la consola
  - Permitir la política de bucket: Concesión de permisos a varias cuentas para cargar objetos o establecer ACL de objetos para el acceso público
  - Denegar la politica de usuarios de IAM de forma explícita: AWS: deniega el acceso a AWS en función de la Región de AWS solicitada
  - Denegar la política de bucket de forma explícita: Requerir SSE-KMS para todos los objetos escritos en un bucket

Configuración de ACL de Amazon S3

Al comprobar la configuración de las ACL, primero debe consultar la configuración de la propiedad del objeto para comprobar si las ACL están habilitadas en el bucket. Tenga en cuenta que los permisos de ACL solo se pueden usar para conceder permisos y no para rechazar solicitudes. Las ACL tampoco se pueden usar para otorgar acceso a los solicitantes rechazados mediante las denegaciones explícitas en las políticas de bucket o de usuarios de IAM.

La configuración de la propiedad del objeto se establece como Aplicada al propietario del bucket

Si la opción Aplicada al propietario del bucket está habilitada, es poco probable que la configuración de las ACL cause un error de acceso denegado (403 Prohibido), ya que esta opción deshabilita todas las ACL que afectan al bucket y a los objetos. Aplicada al propietario del bucket es la configuración predeterminada (y recomendada) para los buckets de Amazon S3.

La opción Aplicada al propietario del bucket se establece como Propietario del bucket preferido o Escritor de objetos

Los permisos de ACL siguen siendo válidos con la configuración Propietario del bucket preferido o Escritor de objetos. Hay dos tipos de ACL: ACL de bucket y ACL de objetos. Para conocer las diferencias entre estos dos tipos de ACL, consulte Mapeo de permisos de ACL y permisos de política de acceso.

Según la acción de la solicitud rechazada, compruebe los permisos de ACL de su bucket o del objeto:

- Si Amazon S3 ha rechazado una `LIST`, un objeto `PUT` o una solicitud `GetBucketAcl` o `PutBucketAcl`, revise los permisos de ACL del bucket.

> **Note**
>
> No puede conceder permisos a objetos `GET` con la configuración de ACL del bucket.

- Si Amazon S3 ha rechazado una solicitud `GET` sobre un objeto de S3 o una solicitud PutObjectAcl, revise los permisos de ACL del objeto.

> **Important**
>
> Si la cuenta propietaria del objeto es diferente a la cuenta propietaria del bucket, entonces la política del bucket no controlará el acceso al objeto.

Solución de un error de acceso denegado (403 Prohibido) provocado por una solicitud de objeto **`GET`** durante la propiedad de un objeto entre cuentas

Revise la Configuración de propiedad de objetos del bucket para determinar el propietario del objeto. Si tiene acceso a las ACL del objeto, también puede comprobar la cuenta del propietario del objeto. (Para ver la cuenta del propietario del objeto, revise la configuración de la ACL del objeto en la consola de Amazon S3). Como alternativa, también puede realizar una solicitud `GetObjectAcl` para encontrar el ID canónico del propietario del objeto a fin de verificar la cuenta del propietario del objeto. De forma predeterminada, las ACL otorgan permisos de forma explícita para las solicitudes `GET` a la cuenta del propietario del objeto.

Tras confirmar que el propietario del objeto es diferente del propietario del bucket, en función de su caso de uso y del nivel de acceso, elija uno de los siguientes métodos para solucionar el error Acceso denegado (403 Prohibido):

- Desactivar las ACL (recomendado): este método se aplicará a todos los objetos y puede ejecutarlo el propietario del bucket. Este método otorga automáticamente la propiedad al propietario del bucket, además de control total sobre cada objeto del bucket. Antes de implementar este método, compruebe los requisitos previos para desactivar las ACL. Para obtener información sobre cómo configurar el bucket como Aplicada al propietario del bucket (recomendado), consulte Configuración de la propiedad de objetos en un bucket existente.

  > **Important**
  >
  > Para evitar un error de acceso denegado (403 Prohibido), asegúrese de migrar los permisos de ACL a una política de bucket antes de desactivar las ACL. Para obtener más información, consulte los ejemplos de políticas de bucket para migrar desde los permisos de ACL.

- Cambiar el propietario del objeto por el propietario del bucket: este método se puede aplicar a objetos individuales, pero solo el propietario del objeto (o un usuario con los permisos adecuados) puede cambiar la propiedad de un objeto. Se pueden aplicar costos de PUT adicionales. (Para obtener más información, consulte Precios de Amazon S3). Este método otorga al propietario del bucket la propiedad total del objeto, lo que le permite controlar el acceso al objeto mediante una política de bucket.

  Para cambiar la propiedad del objeto, lleve a cabo una de las siguientes acciones:

  - Usted (el propietario del bucket) puede volver a copiar el objeto en el bucket.
  - Puede cambiar la configuración de la propiedad del objeto del bucket a la que prefiera el propietario del bucket. Si el control de versiones está desactivado, los objetos del bucket se sobrescriben. Si el control de versiones está activado, aparecerán versiones duplicadas del mismo objeto en el bucket, y el propietario del bucket podrá establecer el vencimiento de una regla de ciclo de vida. Para ver instrucciones sobre cómo cambiar la configuración de Propiedad de objetos, consulte Configuración de la propiedad de objetos en un bucket existente.

> **Note**
>
> Al actualizar la configuración de Propiedad de objetos en la preferida por el propietario del bucket, la configuración solo se aplicará a los objetos nuevos que se carguen en el bucket.

- Puede hacer que el propietario del objeto vuelva a cargar el objeto con la ACL de objetos predefinidos `bucket-owner-full-control`.

> **Note**
>
> Para las cargas entre cuentas, también puede requerir la ACL de objetos predefinidos `bucket-owner-full-control` en la política de bucket. Para ver un ejemplo de política de bucket, consulte Conceder permisos entre cuentas para cargar objetos al mismo tiempo que se garantiza que el propietario del bucket tenga el control total.

- Mantener al escritor de objetos como propietario del objeto: este método no cambia el propietario del objeto, pero sí le permite otorgar acceso a los objetos de forma individual. Para otorgar acceso a un objeto, debe tener el permiso `PutObjectAcl` para el objeto. A continuación, para corregir el error Acceso denegado (403 Prohibido), añada al solicitante como beneficiario para acceder al objeto en las ACL del objeto. Para obtener más información, consulte Configuración de la ACL.

Configuración del bloqueo de acceso público en S3

Si la solicitud que da error implica acceso público o políticas públicas, compruebe la configuración del bloqueo de acceso público en S3 en su cuenta, bucket o punto de acceso de S3. A partir de abril de 2023, todas las configuraciones de bloqueo de acceso público estarán activadas de forma predeterminada para los buckets nuevos. Para obtener más información sobre el concepto de “público” de Amazon S3, consulte Qué significa "pública".

Cuando se establece en `TRUE`, las configuraciones de Bloquear acceso público actúan como políticas de denegación explícitas que anulan los permisos permitidos por las ACL, las políticas de bucket y las políticas de usuarios de IAM. Para determinar si las configuraciones de Bloquear acceso público rechazan su solicitud, revise las siguientes situaciones:

- Si la lista de control de acceso (ACL) es pública, la configuración `BlockPublicAcls` rechazará las llamadas `PutBucketAcl` y `PutObjectACL`.

- Si la solicitud incluye una ACL pública, la configuración `BlockPublicAcls` rechazará las llamadas `PutObject`.
- Si la configuración `BlockPublicAcls` se aplica a una cuenta y la solicitud incluye una ACL pública, cualquier llamada `CreateBucket` que incluya una ACL pública fallará.
- Si el permiso de tu solicitud solo lo concede una ACL pública, la configuración `IgnorePublicAcls` rechazará la solicitud.
- Si la política de bucket especificada permite el acceso público, la configuración `BlockPublicPolicy` rechazará las llamadas `PutBucketPolicy`.
- Si la configuración `BlockPublicPolicy` se aplica a un punto de acceso, fallarán todas las llamadas `PutAccessPointPolicy` y `PutBucketPolicy` que especifiquen una política pública y se realicen a través del punto de acceso.
- Si el punto de acceso o el bucket tienen una política pública, la configuración `RestrictPublicBuckets` rechazará todas las llamadas entre cuentas, excepto las de las entidades principales Servicio de AWS. Esta configuración también rechaza todas las llamadas anónimas (o sin firma).

Para revisar y actualizar las configuraciones de Bloquear acceso público, consulte Establecer la configuración de Block Public Access para sus buckets de S3.

Configuración del cifrado de Amazon S3

Amazon S3 admite el cifrado del lado del servidor en el bucket. El cifrado del lado del servidor es el cifrado de datos en su destino por la aplicación o servicio que los recibe. Amazon S3 cifra sus datos en el nivel de objeto; los escribe en los discos de sus centros de datos de AWS y cuando usted accede a ellos, los descifra para usted.

De forma predeterminada, Amazon S3 aplica ahora el cifrado del lado del servidor con claves administradas por Amazon S3 (SSE-S3) como el nivel básico de cifrado para cada bucket de Amazon S3. Amazon S3 también le permite especificar el método de cifrado del lado del servidor al cargar objetos.

Para revisar el estado de cifrado del lado del servidor y la configuración de cifrado de su bucket

1. Inicie sesión AWS Management Console Management Console y abra la consola de Amazon S3 en https://console.aws.amazon.com/s3/.
2. En el panel de navegación izquierdo, elija Instancias.

3. En la lista Buckets, elija el bucket para el que desea comprobar la configuración de cifrado.
4. Elija la pestaña Propiedades.
5. Desplácese hacia abajo hasta la sección Cifrado predeterminado y consulte la configuración Tipo de cifrado.

Para comprobar la configuración de cifrado mediante la AWS CLI, utilice el comando get-bucket-encryption.

Para comprobar el estado de cifrado de un objeto

1. Inicie sesión AWS Management Console Management Console y abra la consola de Amazon S3 en https://console.aws.amazon.com/s3/.
2. En el panel de navegación izquierdo, elija Instancias.
3. En la lista Buckets, elija el nombre del bucket que contiene el objeto.
4. En la lista Objetos, seleccione el nombre del objeto al que desea agregar cifrado o cuyo cifrado desea modificar.

   Aparece la página de detalles del objeto.
5. Desplácese hacia abajo hasta la sección Configuración del cifrado del lado del servidor para ver la configuración de cifrado del lado del servidor del objeto.

Para comprobar el estado de cifrado del objeto mediante la AWS CLI, utilice el comando head-object.

Requisitos de permisos y cifrado

Amazon S3 admite tres tipos de cifrado del lado del servidor:

- Cifrado en el servidor con claves administradas por Amazon S3 (SSE-S3)
- Cifrado del lado del servidor con claves AWS Key Management Service (AWS KMS) (SSE-KMS)
- Cifrado en el servidor con claves proporcionadas por el cliente (SSE-C)

Según la configuración de cifrado, asegúrese de que se cumplan los siguientes requisitos de permisos:

- SSE-S3: no se requieren permisos adicionales.

- SSE-KMS (con una clave administrada por el cliente): para cargar objetos, se requiere el permiso `kms:GenerateDataKey` en la AWS KMS key. Para descargar objetos y realizar cargas de varias partes, se requiere el permiso `kms:Decrypt` de la clave KMS.
- SSE-KMS (con una Clave administrada de AWS): el solicitante debe ser de la misma cuenta que posee la clave KMS `aws/s3`. El solicitante también debe tener los permisos de Amazon S3 correctos para acceder al objeto.
- SSE-C (con una clave administrada por el cliente): no se requieren permisos adicionales. Puede configurar la política de bucket para requerir y restringir el cifrado del lado del servidor con claves de cifrado proporcionadas por el cliente para los objetos del bucket.

Si el objeto está cifrado con una clave administrada por el cliente, asegúrese de que la política de claves de KMS le permita realizar las acciones `kms:GenerateDataKey` o `kms:Decrypt`. Para obtener instrucciones sobre cómo comprobar la política de claves de KMS, consulte Consultar una política de claves en la Guía para desarrolladores de AWS Key Management Service.

Configuración de bloqueo de objetos de S3

Si el bucket tiene activado el bloqueo de objetos de S3 y el objeto está protegido por un período de retención o una retención legal, Amazon S3 muestra un error de acceso denegado (403 Prohibido) cuando intenta eliminar el objeto.

Para comprobar si el bucket tiene activado el bloqueo de objetos

1. Inicie sesión AWS Management Console Management Console y abra la consola de Amazon S3 en https://console.aws.amazon.com/s3/.
2. En el panel de navegación izquierdo, elija Instancias.
3. En la lista Buckets, elija el nombre del bucket en cuestión.
4. Elija la pestaña Propiedades.
5. Desplácese hacia abajo hasta la sección Bloqueo de objetos. Compruebe si la configuración de Bloqueo de objetos está Habilitada o Deshabilitada.

Para determinar si el objeto está protegido por un período de retención o una retención legal, consulta la información de bloqueo del objeto.

Si el objeto está protegido por un período de retención o una retención legal, compruebe lo siguiente:

- Si la versión del objeto está protegida por el modo de retención de conformidad, no se podrá eliminar permanentemente. Una solicitud `DELETE` permanente de cualquier solicitante, incluido el usuario raíz, generará un error de acceso denegado (403 Prohibido). Además, tenga en cuenta que cuando envía una solicitud `DELETE` para un objeto protegido por el modo de retención de conformidad, Amazon S3 crea un marcador de eliminación para el objeto.
- Si la versión del objeto está protegida con el modo de retención de control y usted tiene el permiso `s3:BypassGovernanceRetention`, puede omitir la protección y eliminar la versión de forma permanente. Para obtener más información, consulte Omitir el modo de gobierno.
- Si la versión del objeto está protegida por una retención legal, una solicitud `DELETE` permanente puede generar un error de acceso denegado (403 Prohibido). Para eliminar la versión del objeto de forma permanente, debe eliminar la retención legal en la versión del objeto. Para eliminar una retención legal, debe tener el permiso `s3:PutObjectLegalHold`. Para obtener más información acerca de la eliminación de una retención legal, consulte Configurar el Bloqueo de objetos de S3.

Política de punto de conexión de VPC

Si accede a Amazon S3 desde un punto de conexión de nube privada virtual (VPC), asegúrese de que la política de punto de conexión de VPC no le impida acceder a los recursos de Amazon S3. De forma predeterminada, la política de puntos de conexión de VPC permite todas las solicitudes a Amazon S3. También puedes configurar la política de puntos de conexión de VPC para restringir determinadas solicitudes. Para obtener información sobre cómo comprobar la política de puntos de conexión de VPC, consulte Uso de políticas de punto de conexión para controlar el acceso a puntos de conexión de VPC en la Guía de AWS PrivateLink.

Políticas de AWS Organizations

Si su Cuenta de AWS pertenece a una organización, las políticas de AWS Organizations pueden impedirle acceder a los recursos de Amazon S3. Las políticas de AWS Organizations no bloquean ninguna solicitud a Amazon S3 de forma predeterminada. Sin embargo, asegúrese de que sus políticas de AWS Organizations no estén configuradas para bloquear el acceso a los buckets de S3. Para obtener instrucciones sobre cómo comprobar sus políticas de AWS Organizations, consulte Enumeración de todas las políticas en la Guía del usuario de AWS Organizations.

Configuración del punto de acceso

Si recibe un error de acceso denegado (403 Prohibido) al realizar solicitudes a través de los puntos de acceso de Amazon S3, es posible que tenga que comprobar lo siguiente:

- Las configuraciones de los puntos de acceso
- La política de usuarios de IAM que se utiliza para los puntos de acceso
- La política de bucket que se usa para administrar o configurar los puntos de acceso entre cuentas

Configuraciones y políticas de los puntos de acceso

- Al crear un punto de acceso, el origen de la red puede ser Internet o VPC. Si el origen de la red se establece solo en VPC, Amazon S3 rechazará cualquier solicitud realizada al punto de acceso que no provenga de la VPC especificada. Para comprobar el origen de la red del punto de acceso, consulte Crear puntos de acceso restringidos a una nube privada virtual.
- Con los puntos de acceso, también se puede personalizar la configuración del bloqueo de acceso público, que es similar a la configuración del bloqueo de acceso público en el nivel de bucket o de cuenta. Para ver la configuración personalizada de Bloquear acceso público, consulte Administrar el acceso público a los puntos de acceso.
- Para realizar solicitudes correctas a Amazon S3 a través de los puntos de acceso, asegúrese de que el solicitante tenga los permisos de IAM necesarios. Para obtener más información, consulte Configurar las políticas de IAM para el uso de puntos de acceso.
- Si la solicitud incluye puntos de acceso entre cuentas, asegúrese de que el propietario del bucket haya actualizado la política del bucket para autorizar las solicitudes desde el punto de acceso. Para obtener más información, consulte Concesión de permisos para puntos de acceso entre cuentas.

Si el error Acceso denegado (403 Prohibido) persiste después de consultar todos los elementos de este tema, obtenga el ID de solicitud de Amazon S3 y póngase en contacto con AWS Support para obtener más información.

Solución de problemas de operaciones por lotes

Los siguientes temas describen errores comunes para ayudarlo a solucionar problemas que pudieran surgir durante las operaciones por lotes.

Errores comunes

- El informe de trabajo no se entrega cuando hay un problema de permisos o está activado un modo de retención de bloqueo de objetos de S3

- Error de replicación por lotes de S3 con error: La generación del manifiesto no ha encontrado claves que coincidan con los criterios del filtro
- Los errores en las operaciones por lotes se producen después de agregar una nueva regla de replicación a una configuración de replicación existente
- Objetos de las operaciones por lotes que fallan con el error 400 InvalidRequest: Error en la tarea porque falta VersionId
- Error al crear un trabajo con la opción de etiqueta de trabajo activada
- Acceso denegado para leer el manifiesto

El informe de trabajo no se entrega cuando hay un problema de permisos o está activado un modo de retención de bloqueo de objetos de S3

El siguiente error se produce si faltan los permisos necesarios o si el modo de retención de bloqueo de objetos (ya sea el modo de gobernanza o el modo de cumplimiento) está activado en el bucket de destino.

Error: Motivos del error. El informe del trabajo no se ha podido escribir en el bucket del informe. Compruebe sus permisos.

El rol de IAM y la política de confianza deben configurarse para permitir que las operaciones por lotes de S3 accedan a los objetos PUT del bucket donde se entregará el informe. Si faltan estos permisos necesarios, se produce un error en la entrega del informe de trabajo.

Si hay un modo de retención activado, el bucket está protegido con escritura única y lectura múltiple (WORM). No se admite el bloqueo de objetos con el modo de retención activado en el bucket de destino, por lo que los intentos de entrega del informe de finalización de trabajo fallarán. Para solucionar este problema, elija un bucket de destino para los informes de finalización de trabajos que no tenga activado el modo de retención de bloqueo de objetos.

Error de replicación por lotes de S3 con error: La generación del manifiesto no ha encontrado claves que coincidan con los criterios del filtro

Error: La generación del manifiesto no ha encontrado claves que coincidan con los criterios del filtro.

Este error se produce por las razones siguientes:

- Cuando los objetos del bucket de origen se almacenan en la clase de almacenamiento S3 Glacier Flexible Retrieval o S3 Glacier Deep Archive.

Para utilizar la replicación por lotes en estos objetos, primero debe restaurar la clase de almacenamiento S3 Standard mediante una operación de inicio de restauración de objetos de S3 en un trabajo de operaciones por lote. Para obtener más información, consulte Restauración de un objeto archivado y Restaurar objetos (operaciones por lotes). Una vez restaurados los objetos, puede replicarlos con un trabajo de replicación por lotes.

- Si los criterios de filtro proporcionados no coinciden con ningún objeto válido del bucket de origen.

  Compruebe y corrija los criterios del filtro. Por ejemplo, en la regla de replicación por lotes, el criterio de filtrado busca todos los objetos en el*amzn-s3-demo-bucket* que tienen el prefijo `Tax/`. Si el nombre del prefijo se ha introducido de forma incorrecta, con una barra diagonal al principio y al final de `/Tax/` en lugar de solo al final, no se encontrará ningún objeto de S3. Para resolver el error, corrija el prefijo, en este caso, de `/Tax/` a `Tax/` en la regla de replicación.

Los errores en las operaciones por lotes se producen después de agregar una nueva regla de replicación a una configuración de replicación existente

Las operaciones por lotes intentan realizar la replicación de objetos existentes para cada regla de la configuración de replicación del bucket de origen. Si hay problemas con alguna de las reglas de replicación existentes, pueden producirse errores.

En el informe de finalización del trabajo de operaciones por lotes se explican los motivos del error del trabajo. Para obtener una lista de los errores comunes, consulte Motivos de errores de replicación de Amazon S3.

Objetos de las operaciones por lotes que fallan con el error 400 InvalidRequest: Error en la tarea porque falta VersionId

El siguiente ejemplo de error se produce si un trabajo de operaciones por lote realiza acciones en objetos de un bucket versionado y encuentra un objeto en el manifiesto con un campo de ID de versión vacío.

Error: *BUCKET_NAME,prefix/file_name*,failed,400,InvalidRequest,Task failed due to missing VersionId

Este error se produce porque el campo ID de versión del manifiesto es una cadena vacía, en lugar de la cadena `null` literal.

Las operaciones por lotes fallarán para estos objetos en particular, pero no para todo el trabajo. Este problema se produce si el formato del manifiesto está configurado para usar ID de versión durante la operación. Los trabajos no versionados no generan este problema porque solo funcionan en la versión más reciente de cada objeto e ignoran los ID de versión del manifiesto.

Para solucionar este problema, convierta los ID de versión vacíos en cadenas `null`. Para obtener más información, consulte the section called “Convertir cadenas de ID de versión vacías en cadenas nulas”.

Error al crear un trabajo con la opción de etiqueta de trabajo activada

Sin el permiso `s3:PutJobTagging`, crear trabajos de operaciones por lotes con la opción de etiqueta de trabajo activada genera errores `403 access denied`.

Para crear trabajos de operaciones por lotes con la opción de etiqueta de trabajo activada, el usuario (de IAM) AWS Identity and Access Management que crea el trabajo de operaciones por lotes debe tener el permiso `s3:PutJobTagging` además del permiso `s3:CreateJob`.

Para obtener más información sobre los permisos necesarios para las operaciones por lotes, consulte the section called “Concesión de permisos”.

Acceso denegado para leer el manifiesto

Si las operaciones por lotes no pueden leer el archivo de manifiesto al intentar crear un trabajo de operaciones por lotes, pueden producirse los siguientes errores.

AWS CLI

Motivo del error Lectura del manifiesto prohibida: AccessDenied

Consola de Amazon S3

Aviso: No se ha podido obtener la ETag del objeto del manifiesto. Especifique un objeto distinto para continuar.

Para resolver este problema, siga uno de estos pasos:

- Compruebe que el rol de IAM para la Cuenta de AWS que ha utilizado para crear el trabajo de operaciones por lotes tenga permisos `s3:GetObject`. El rol de IAM de la cuenta debe tener permisos `s3:GetObject` para permitir que las operaciones por lotes lean el archivo de manifiesto.

Para obtener más información sobre los permisos necesarios para las operaciones por lotes, consulte the section called “Concesión de permisos”.

- Compruebe los metadatos de los objetos del manifiesto para ver si hay algún desajuste de acceso con la propiedad de objetos de S3. Para obtener más información acerca de la propiedad de objetos de S3, consulte the section called “Control de la propiedad de objetos”.
- Compruebe si se utilizan claves de AWS Key Management Service (AWS KMS) para cifrar el archivo de manifiesto.

  La herramienta de operaciones por lotes de es compatible con los informes de inventario CSV cifrados con AWS KMS. Sin embargo, las operaciones por lotes no admiten archivos de manifiesto CSV cifrados con AWS KMS. Para obtener más información, consulte Configuración de Inventario de Amazon S3 y Especificar un manifiesto.

Solucionar problemas de Amazon S3 Lifecycle

La siguiente información puede ayudarle a solucionar problemas habituales con las reglas de Amazon S3 Lifecycle.

Temas

He ejecutado una operación de lista en mi bucket y he visto objetos que pensaba que habían caducado o cambiado de conformidad con una regla del ciclo de vida.

Las transiciones de objetos y los vencimientos de los objetos de S3 Lifecycle son operaciones asincrónicas. Por lo tanto, es posible que haya un retraso entre el momento en que los objetos cumplan los requisitos de caducidad o cambio y el momento en que realmente estén cambiando o caduquen. Los cambios en la facturación se aplican en cuanto se cumple la regla de ciclo de vida, incluso aunque la acción no se haya completado. La excepción a este comportamiento es si hay una regla de ciclo de vida para pasar a la clase de almacenamiento S3 Intelligent-Tiering. En ese caso, los cambios en la facturación no se producen hasta que el objeto haya pasado a S3 Intelligent-Tiering. Para obtener más información sobre los cambios en la facturación, consulte Configurar el ciclo de vida de un bucket.

> **Note**
>
> Amazon S3 no realiza la transición de objetos de menos 128 KB de la clase de almacenamiento S3 Standard o S3 Standard-IA a la clase de almacenamiento S3 Intelligent-Tiering, S3 Standard-IA o S3 One Zone-IA.

¿Cómo puedo supervisar las acciones que se llevan a cabo según mis reglas de ciclo de vida?

Para supervisar las acciones que se llevan a cabo según las reglas de ciclo de vida, puede utilizar las siguientes características:

- Notificaciones de eventos de S3: puede configurar las notificaciones de eventos de S3 para que se le notifique cualquier evento de transición o caducidad del ciclo de vida de S3.
- Registros de acceso al servidor de S3: puede habilitar los registros de acceso al servidor para los buckets de S3 para capturar acciones del ciclo de vida de S3, como, por ejemplo, las transiciones de objetos a otra clase de almacenamiento o la caducidad de objetos. Para obtener más información, consulte Ciclos de vida y registros.

Para ver los cambios en el almacenamiento provocados por las acciones del ciclo de vida a diario, le recomendamos que utilice los paneles de Almacenamiento de lente de S3 en lugar de utilizar las

métricas de Amazon CloudWatch. En el panel de Almacenamiento de lente puede ver las siguientes métricas, que supervisan el número o el tamaño de los objetos:

- Bytes de la versión actual
- Recuento de objetos de la versión actual
- Bytes de la versión que no es actual
- Recuento de objetos de la versión que no es actual
- Recuento de objetos del marcador de eliminación
- Bytes de almacenamiento del marcador de eliminación
- Bytes de carga multiparte incompletos
- Recuento de objetos con carga multiparte incompleta

Mi recuento de objetos de S3 sigue aumentando, incluso después de configurar reglas de ciclo de vida en un bucket con control de versiones activado.

En un bucket con control de versiones, cuando un objeto caduca, este no se elimina por completo del bucket. En su lugar, se crea un marcador de eliminación como la versión más reciente del objeto. Los marcadores de eliminación también cuentan como objetos. Por lo tanto, si se crea una regla de ciclo de vida para que caduquen solo las versiones actuales, el recuento de objetos en el bucket de S3 en realidad aumenta en lugar de disminuir.

Por ejemplo, supongamos que un bucket de S3 tiene activado el control de versiones con 100 objetos y que una regla de ciclo de vida está configurada para que las versiones actuales del objeto caduquen transcurridos 7 días. Después del séptimo día, el recuento de objetos aumenta a 200 porque se crean 100 marcadores de eliminación además de los 100 objetos originales, que ahora son las versiones no actuales. Para obtener más información sobre las acciones de las reglas de configuración de S3 Lifecycle para los buckets con control de versiones, consulte Configurar el ciclo de vida de un bucket.

Para eliminar objetos de forma permanente, añada una configuración de ciclo de vida adicional para eliminar las versiones anteriores de los objetos, los marcadores de eliminación caducados y las cargas multiparte incompletas. Para obtener instrucciones sobre cómo crear nuevas reglas de ciclo de vida, consulte Configurar el ciclo de vida de un bucket.

Note

- Amazon S3 redondea la fecha de transición o caducidad de un objeto a la medianoche UTC del día siguiente.

  Al evaluar los objetos para las acciones del ciclo de vida, Amazon S3 utiliza la hora de creación del objeto en UTC. Por ejemplo, supongamos que tenemos un bucket sin control de versiones con una regla de ciclo de vida configurada para que los objetos caduquen al cabo de un día. Supongamos que un objeto se creó el 1 de enero a las 17:05 PDT, que corresponde al 2 de enero a las 00:05 UTC. El objeto tiene un día de antigüedad a las 00:05 UTC del 3 de enero, por lo que podrá caducar cuando S3 Lifecycle evalúe los objetos a las 00:00 UTC del 4 de enero.

  Dado que las acciones del ciclo de vida de Amazon S3 se producen de forma asíncrona, es posible que haya algún retraso entre la fecha especificada en la regla del ciclo de vida y la transición física real del objeto. Para obtener más información, consulte Retraso de caducidad o transición.

  Para obtener más información, consulte Reglas de ciclo de vida: basadas en la edad de un objeto
- En el caso de los objetos de S3 que están protegidos por un bloqueo de objetos, las versiones actuales no se eliminan de forma permanente. En su lugar, se añade un marcador de eliminación a los objetos, por lo que dejan de ser actuales. Las versiones no actuales se conservan y no caducan de forma permanente.

¿Cómo puedo vaciar mi bucket de S3 con las reglas de ciclo de vida?

Las reglas de ciclo de vida de S3 son una herramienta eficaz para vaciar un bucket de S3 con millones de objetos. Para eliminar una gran cantidad de objetos del bucket de S3, asegúrese de utilizar estos dos pares de reglas de ciclo de vida:

- Hacer que caduquen las versiones actuales de los objetos y Eliminar de forma permanente las versiones anteriores de los objetos
- Eliminar los marcadores de eliminación caducados y Eliminar las cargas multiparte incompletas

Para obtener instrucciones sobre cómo crear nuevas reglas de configuración del ciclo de vida, consulte Configurar el ciclo de vida de un bucket.

> **Note**
>
> En el caso de los objetos de S3 que están protegidos por un bloqueo de objetos, las versiones actuales no se eliminan de forma permanente. En su lugar, se añade un marcador de eliminación a los objetos, por lo que dejan de ser actuales. Las versiones no actuales se conservan y no caducan de forma permanente.

Mi factura de Amazon S3 ha aumentado tras pasar los objetos a una clase de almacenamiento más barata.

Hay varios motivos por los que la factura podría aumentar tras cambiar los objetos a una clase de almacenamiento más barata:

- Gastos generales de S3 Glacier para objetos pequeños

  Por cada objeto que pasa a S3 Glacier Flexible Retrieval o S3 Glacier Deep Archive Flexible Retrieval, esta actualización de almacenamiento tiene asociada un gasto adicional total de 40 KB. Dentro del gasto adicional de 40 KB, se utilizan 8 KB para almacenar los metadatos y el nombre del objeto. Estos 8 KB se cobran según las tarifas de S3 Standard. Los 32 KB restantes se utilizan para la indexación y los metadatos relacionados. A estos 32 KB se les aplican los precios de S3 Glacier Flexible Retrieval o S3 Glacier Deep Archive.

  Por lo tanto, si almacena muchos objetos de menor tamaño, no recomendamos utilizar transiciones del ciclo de vida. En su lugar, para reducir los cargos adicionales, le recomendamos agregar muchos objetos pequeños en un número menor de objetos grandes antes de almacenarlos en Amazon S3. Para obtener más información sobre las consideraciones de costos, consulte Transición a las clases de almacenamiento S3 Glacier Flexible Retrieval y S3 Glacier Deep Archive (archivo de objetos).
- Gastos de almacenamiento mínimos

  Algunas clases de almacenamiento de S3 tienen requisitos mínimos de duración de almacenamiento. A los objetos que se eliminen, sobrescriban o cambien de esas clases antes de cumplir con la duración mínima se les cobrará una tarifa de transición o eliminación anticipada prorrateada. Estos requisitos mínimos de duración del almacenamiento son los siguientes:

- S3 Standard-IA y S3 One Zone-IA: 30 días
- S3 Glacier Flexible Retrieval y S3 Glacier Instant Retrieval: 90 días
- S3 Glacier Deep Archive: 180 días

Para obtener más información sobre estos requisitos, consulte la sección Restricciones de Transición de objetos con Amazon S3 Lifecycle. Para obtener información general sobre los precios de S3, consulte Precios de Amazon S3 y la calculadora de precios de AWS.

- Costos de la transición del ciclo de vida

  Cada vez que un objeto pasa a una clase de almacenamiento diferente según la regla de ciclo de vida, Amazon S3 contabiliza dicha transición como una solicitud de transición. Los costos de estas solicitudes de transición se suman a los costos de esas clases de almacenamiento. Si tiene previsto pasar una gran cantidad de objetos, tenga en cuenta los costos de la solicitud al pasar a un nivel inferior. Para obtener más información, consulte Precios de Amazon S3.

He actualizado mi política de bucket, pero las reglas del ciclo de vida caducadas siguen borrando mis objetos de S3.

Las instrucciones `Deny` de una política de bucket no impiden que caduquen los objetos definidos en una regla de ciclo de vida. Las acciones del ciclo de vida (como las transiciones o los vencimientos) no utilizan la operación `DeleteObject` de S3. En cambio, las acciones del ciclo de vida de S3 se realizan mediante puntos de conexión internos de S3. (Para obtener más información, consulte Ciclos de vida y registros).

Para evitar que la regla del ciclo de vida lleve a cabo alguna acción, debe editar, eliminar o desactivar la regla.

¿Puedo recuperar objetos de S3 que hayan caducado según las reglas de S3 Lifecycle?

La única forma de recuperar los objetos caducados de conformidad con S3 Lifecycle es mediante el control de versiones, que debe establecerse antes de que los objetos cumplan los requisitos de caducidad. No puede deshacer las operaciones de caducidad que realicen las reglas del ciclo de vida. Si las reglas de S3 Lifecycle eliminan los objetos de forma permanente, no podrá recuperarlos. Para activar el control de versiones en un bucket, consulte the section called “Usar S3 Versioning”.

Si ha aplicado el control de versiones al bucket y las versiones no actuales de los objetos siguen intactas, puede restaurar las versiones anteriores de los objetos caducados. Para obtener más información sobre el comportamiento de las acciones de las reglas y los estados de control de versiones de S3 Lifecycle, consulte la tabla Acciones del ciclo de vida y estado de las versiones del bucket en Elementos para describir las acciones del ciclo de vida.

> **Note**
>
> Si el bucket de S3 está protegido por Copia de seguridad de AWS o Replicación de objetos de S3, es posible que también pueda utilizar estas funciones para recuperar los objetos caducados.

¿Cómo puedo excluir un prefijo de la regla de ciclo de vida?

S3 Lifecycle no admite la exclusión de prefijos en sus reglas. En su lugar, utilice etiquetas para etiquetar todos los objetos que desee incluir en la regla. Para obtener más información sobre cómo usar etiquetas en reglas de ciclo de vida, consulte the section called “Ejemplo 1: especificar un filtro”.

¿Cómo puedo incluir varios prefijos en la regla de ciclo de vida?

S3 Lifecycle no permite incluir varios prefijos en sus reglas. En su lugar, utilice etiquetas para etiquetar todos los objetos que desee incluir en la regla. Para obtener más información sobre cómo usar etiquetas en reglas de ciclo de vida, consulte the section called “Ejemplo 1: especificar un filtro”.

Sin embargo, si tiene uno o más prefijos que comienzan con los mismos caracteres, puede incluir todos esos prefijos en la regla especificando un prefijo parcial sin barra al final (/) en el filtro. Por ejemplo, supongamos que dispone de los siguientes prefijos:

```
sales1999/
sales2000/
sales2001/
```

Para incluir los tres prefijos en la regla, especifique `<Prefix>sales</Prefix>` en la regla de ciclo de vida.

Solución de problemas de replicación

En esta sección se incluyen consejos de solución de problemas para Replicación de Amazon S3 e información sobre los errores de la replicación por lotes de S3.

Temas

Consejos para solucionar problemas de replicación de S3

Si las réplicas de objetos no aparecen en el bucket de destino después de configurar la replicación, use estos consejos de solución de problemas para identificar y solucionar los problemas.

- La mayoría de los objetos se replican en 15 minutos. El tiempo que tarda Amazon S3 en replicar un objeto depende de diferentes factores, como el par de regiones de origen y destino y el tamaño del objeto. La replicación puede tardar varias horas para los objetos grandes. Para obtener visibilidad de los tiempos de replicación, puede utilizar el control del tiempo de replicación de S3 (S3 RTC).

  Si el objeto que se replica es grande, espere un tiempo antes de comprobar si aparece en el destino. También puede comprobar el estado de replicación del objeto de origen. Si el estado de replicación de objetos es `PENDING`, Amazon S3 no ha completado la replicación. Si el estado de replicación del objeto es `FAILED`, compruebe la configuración de replicación establecida en el bucket de origen. Además, para recibir información sobre los errores durante la replicación, puede configurar la replicación de las notificaciones de eventos de Amazon S3 para recibir eventos de error. Para obtener más información, consulte Recepción de eventos de error de replicación con notificaciones de eventos de Amazon S3.
- Puede llamar a la operación de la API `HeadObject` para comprobar el estado de replicación de un objeto. La operación de la API `HeadObject` devuelve el estado de replicación `PENDING`, `COMPLETED` o `FAILED` de un objeto. En respuesta a una llamada a la API `HeadObject`, el estado de replicación se devuelve en el elemento `x-amz-replication-status`.

  > **Note**
  >
  > Para ejecutar `HeadObject`, debe tener acceso de lectura al objeto que solicita. Una solicitud `HEAD` tiene las mismas opciones que una solicitud `GET`, sin realizar ninguna

operación `GET`. Por ejemplo, para ejecutar una solicitud `HeadObject` mediante la AWS Command Line Interface (AWS CLI), puede ejecutar el siguiente comando. Reemplace los *`user input placeholders`* con su propia información.

```
aws s3api head-object --bucket my-bucket --key index.html
```

- Después de que `HeadObject` devuelva los objetos con un estado de replicación `FAILED`, puede utilizar la replicación por lotes de S3 para replicar esos objetos con errores. Como alternativa, puede volver a cargar los objetos con errores en el bucket de origen, lo que iniciará la replicación de los objetos nuevos.
- En la configuración de replicación en el bucket de origen, verifique lo siguiente:
  - El Nombre de recurso de Amazon (ARN) del bucket de destino es correcto.
  - El prefijo de nombre de clave sea correcto. Por ejemplo, si establece la configuración para replicar objetos con el prefijo `Tax`, entonces, solo se replicarán los objetos con nombres de clave como `Tax/document1` o `Tax/document2`. No se replicará un objeto con el nombre de clave `document3`.
  - El estado de la regla de replicación es `Enabled`.
- Compruebe que el control de versiones no se haya suspendido en ningún bucket de la configuración de la replicación. Ambos buckets de origen y destino deben tener habilitado el control de versiones.
- Si una regla de replicación está configurada como Cambiar la propiedad de los objetos al propietario del bucket de destino, el rol (de IAM) AWS Identity and Access Management que se usa para la replicación debe tener el permiso `s3:ObjectOwnerOverrideToBucketOwner`. Este permiso se concede al recurso (en este caso, al bucket de destino). Por ejemplo, la siguiente instrucción `Resource` muestra cómo conceder este permiso en el bucket de destino:

```
{
  "Effect":"Allow",
  "Action":[
    "s3:ObjectOwnerOverrideToBucketOwner"
  ],
  "Resource":"arn:aws:s3:::DestinationBucket/*"
}
```

- Si el bucket de destino pertenece a otra cuenta, el propietario del bucket de destino también debe conceder el permiso `s3:ObjectOwnerOverrideToBucketOwner` al propietario del bucket de

origen mediante la política de bucket de destino. Para utilizar el siguiente ejemplo de política de bucket, sustituya *user input placeholders* con su información:

```
{
  "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 la configuración de la propiedad de los objetos incluye Aplicada al propietario del bucket, no será necesario que actualice la configuración a Cambiar la propiedad de los objetos al propietario del bucket de destino en la regla de replicación. El cambio de propiedad de los objetos se realiza de forma predeterminada. Para obtener más información acerca de cómo cambiar la propiedad de la réplica, consulte Cambiar el propietario de la réplica.

- Si va a definir la configuración de replicación en un escenario entre cuentas en el que los buckets de origen y destino pertenecen a diferentes Cuentas de AWS, los buckets de destino no se podrán configurar como buckets de Pago por solicitante. Para obtener más información, consulte Utilización de buckets de pago por solicitante para transferencias de almacenamiento y uso.
- Si los objetos de origen de un bucket están cifrados con una clave de AWS Key Management Service (AWS KMS), la regla de replicación debe configurarse para incluir objetos cifrados con AWS KMS. Asegúrese de seleccionar Replicar objetos cifrados con AWS KMS en la configuración

Cifrado de la consola de Amazon S3. A continuación, seleccione una clave de AWS KMS para cifrar los objetos de destino.

> **Note**
>
> Si el bucket de destino está en una cuenta diferente, especifique una clave administrada por el cliente de AWS KMS que pertenezca a la cuenta de destino. No utilice la clave administrada predeterminada de Amazon S3 (`aws/s3`). Al usar la clave predeterminada se cifran los objetos con la clave administrada de Amazon S3 que pertenece a la cuenta de origen, lo que impide que el objeto se comparta con otra cuenta. Como resultado, la cuenta de destino no podrá acceder a los objetos del bucket de destino.

Para utilizar una clave de AWS KMS que pertenezca a la cuenta de destino para cifrar los objetos de destino, la cuenta de destino debe conceder los permisos `kms:GenerateDataKey` y `kms:Encrypt` al rol de replicación de la política de claves de KMS. Para utilizar el siguiente ejemplo de instrucción en su política de claves de KMS, sustituya *`user input placeholders`* con su información:

```
{
    "Sid": "AllowS3ReplicationSourceRoleToUseTheKey",
    "Effect": "Allow",
    "Principal": {
        "AWS": "arn:aws:iam::123456789101:role/s3-replication-role"
    },
    "Action": ["kms:GenerateDataKey", "kms:Encrypt"],
    "Resource": "*"
}
```

Si usa un asterisco (*) para la instrucción `Resource` de la política de claves de AWS KMS, la política otorga permiso para usar la clave de KMS únicamente para el rol de replicación. La política no permite que el rol de replicación derive sus permisos.

De forma predeterminada, la política de claves de KMS otorga al usuario raíz todos los permisos sobre la clave. Estos permisos se pueden delegar a otros usuarios de la misma cuenta. A menos que haya instrucciones `Deny` en la política de claves de KMS de origen, basta con utilizar una política de IAM para conceder permisos de rol de replicación a la clave de KMS de origen.

> **Note**
>
> Las políticas de claves de KMS que restringen el acceso a rangos de CIDR, puntos de conexión de VPC o puntos de acceso de S3 específicos pueden provocar un error en la replicación.

Si las claves KMS de origen o destino otorgan permisos en función del contexto de cifrado, confirme que las claves de bucket de Amazon S3 estén activadas para los buckets. Si los buckets tienen activadas las claves de bucket de S3, el contexto de cifrado debe ser el recurso en el nivel de bucket, tal como se muestra a continuación:

```
"kms:EncryptionContext:arn:aws:arn": [
     "arn:aws:s3:::SOURCE_BUCKET_NAME"
     ]
"kms:EncryptionContext:arn:aws:arn": [
     "arn:aws:s3:::DESTINATION_BUCKET_NAME"
     ]
```

Además de los permisos otorgados por la política de claves de KMS, la cuenta de origen debe añadir los siguientes permisos mínimos a la política de IAM del rol de replicación:

```
{
    "Effect": "Allow",
    "Action": [
        "kms:Decrypt",
        "kms:GenerateDataKey"
    ],
    "Resource": [
        "SourceKmsKeyArn"
    ]
},
{
    "Effect": "Allow",
    "Action": [
        "kms:GenerateDataKey",
        "kms:Encrypt"
    ],
    "Resource": [
        "DestinationKmsKeyArn"
```

```
    ]
}
```

Para obtener más información sobre la replicación de objetos cifrados con AWS KMS, consulte Replicar objetos cifrados.

- Si el bucket de destino pertenece a otra Cuenta de AWS, compruebe que el propietario del bucket tenga una política de bucket en el bucket de destino que permita al propietario del bucket de origen replicar objetos. Para ver un ejemplo, consulte La configuración de la reproducción para los buckets de origen y destino son propiedad de diferentes cuentas.
- Si los objetos siguen sin replicarse después de haber validado los permisos, compruebe si hay instrucciones `Deny` explícitas en las siguientes ubicaciones:
  - Las instrucciones `Deny` en las políticas de bucket de origen o destino. La replicación devuelve un error si la política de bucket deniega el acceso a la función de replicación para cualquiera de las siguientes acciones:

    Bucket de origen:

    ```
            "s3:GetReplicationConfiguration",
            "s3:ListBucket",
            "s3:GetObjectVersionForReplication",
            "s3:GetObjectVersionAcl",
            "s3:GetObjectVersionTagging"
    ```

    Buckets de destino:

    ```
            "s3:ReplicateObject",
            "s3:ReplicateDelete",
            "s3:ReplicateTags"
    ```

  - Las instrucciones `Deny` o límites de permisos adjuntos al rol de IAM pueden provocar un error en la replicación.
  - Las instrucciones `Deny` en las políticas de control de servicios de AWS Organizations adjuntas a las cuentas de origen o destino pueden provocar un error en la replicación.

- Si la réplica de un objeto no aparece en el bucket de destino, los siguientes problemas podría haber impedido la replicación:
  - Amazon S3 no replica los objetos de un bucket de origen si son una réplica creada por otra configuración de replicación. Por ejemplo, si establece la configuración de replicación del bucket A en el bucket B y, luego, en el bucket C, Amazon S3 no replica las réplicas de objetos del bucket B en el bucket C.
  - Un propietario del bucket de origen puede conceder permisos a otras Cuentas de AWS para cargar objetos. De forma predeterminada, el propietario del bucket de origen no tiene permisos sobre los objetos creados por otras cuentas. La configuración de replicación solo replica los objetos para los que el propietario del bucket de origen tiene permisos de acceso. El propietario del bucket de origen puede conceder permisos a otras Cuentas de AWS para crear objetos con la condición de que tengan permisos de acceso explícitos para esos objetos. Para ver una política de ejemplo, consulte Conceder permisos entre cuentas para cargar objetos al mismo tiempo que se garantiza que el propietario del bucket tenga el control total.
- Supongamos que en la configuración de replicación añade una regla para replicar un subconjunto de objetos con una etiqueta específica. En este caso, debe asignar la clave de etiqueta y el valor específicos en el momento de crear el objeto para que Amazon S3 replique el objeto. Si primero crea un objeto y luego agrega la etiqueta al objeto existente, Amazon S3 no replica el objeto.
- Use las notificaciones de eventos de Amazon S3 para recibir notificaciones de instancias cuando los objetos no se repliquen en la Región de AWS de destino. Las notificaciones de eventos de Amazon S3 están disponibles a través de Amazon Simple Queue Service (Amazon SQS), Amazon Simple Notification Service (Amazon SNS) o AWS Lambda. Para obtener más información, consulte Recepción de eventos de error de replicación con notificaciones de eventos de Amazon S3.

  También puede usar las notificaciones de eventos de Amazon S3 para ver los motivos de error de replicación. Para revisar la lista de los motivos de error, consulte Motivos de errores de replicación de Amazon S3.

Errores de replicación por lotes

Para solucionar problemas con objetos que no se replican en el bucket de destino, compruebe los diferentes tipos de permisos del bucket, el rol de replicación y el rol de IAM que se utilizan para crear el trabajo de replicación por lotes. Además, compruebe la configuración de acceso público y la configuración de propiedad del bucket.

Al utilizar la replicación por lotes, es posible que se produzca uno de los siguientes errores:

- Error en el estado de operación por lotes por el motivo siguiente: no se ha podido escribir el informe del trabajo en el bucket de informes.

  Este error se produce si el rol de IAM que se utiliza para el trabajo de operaciones por lotes no puede colocar el informe de finalización en la ubicación que se especificó al crear el trabajo. Para resolver este error, compruebe que el rol de IAM tenga permisos `PutObject` para el bucket en el que desea guardar el informe de finalización de las operaciones por lotes. Se recomienda entregar el informe en un bucket diferente al bucket de origen.

- La operación por lotes se ha completado con errores y el total de errores no es igual a 0.

  Este error se produce si no hay problemas porque no hay suficientes permisos de objetos con el trabajo de replicación por lotes que se está ejecutando. Si utiliza una regla de replicación para su trabajo de replicación por lotes, asegúrese de que el rol de IAM utilizado para la replicación tenga los permisos adecuados para acceder a los objetos del bucket de origen o de destino. También puede consultar el Informe de finalización de replicación por lotes para revisar el Motivo de error de replicación de Amazon S3.

- El trabajo por lotes se ha ejecutado correctamente, pero el número de objetos esperado en el bucket de destino no es el mismo.

  Este error se produce cuando hay una discrepancia entre los objetos que figuran en el manifiesto que se proporciona en el trabajo de replicación por lotes y los filtros que ha seleccionado al crear el trabajo. También puede recibir este mensaje cuando los objetos de su bucket de origen no coincidan con ninguna regla de replicación y no estén incluidos en el manifiesto generado.

Solucionar problemas de registro de acceso al servidor

Los siguientes temas pueden ayudarlo a solucionar problemas que pueden surgir al configurar el registro con Amazon S3.

Temas

- Mensajes de error comunes al configurar el registro
- Solución de los errores de entrega

Mensajes de error comunes al configurar el registro

Al habilitar el registro a través de la AWS Command Line Interface (AWS CLI) y los SDK de AWS, pueden aparecer los siguientes mensajes de error:

Error: Registro entre ubicaciones de S3 no permitido

Si el bucket de destino se encuentra en una región diferente a la del bucket de origen, se produce un error de Registro entre ubicaciones de S3 no permitido. Para resolver este error, asegúrese de que el bucket de destino configurado para recibir los registros de acceso esté en la misma Región de AWS y Cuenta de AWS que el bucket de origen.

Error: El propietario del bucket que se va a registrar y el bucket de destino deben ser iguales

Al habilitar el registro de acceso al servidor, este error se produce si el bucket de destino especificado pertenece a una cuenta diferente. Para resolver este error, asegúrese de que el bucket de destino esté en la misma Cuenta de AWS que el bucket de origen.

> **Note**
>
> Le recomendamos que elija un bucket de destino diferente al bucket de origen. Cuando los buckets de origen y destino son el mismo, se crean registros adicionales para los registros que se escriben en el bucket, lo que puede incrementar la factura de almacenamiento. Estos registros adicionales sobre los registros también pueden dificultar la búsqueda de los registros concretos que está buscando. Para que la administración de registros sea más sencilla, le recomendamos que guarde los registros de acceso en un bucket distinto. Para obtener más información, consulte the section called “¿Cómo habilito la entrega de registros?”.

Error: El bucket de destino para el registro no existe

El bucket de destino debe existir antes de definir la configuración. Este error indica que el bucket de destino no existe o no se encuentra. Asegúrese de el nombre del bucket esté bien escrito y, a continuación, vuelva a intentarlo.

Error: Concesiones de destino no permitidas para los buckets aplicados al propietario del bucket

Este error indica que el bucket de destino utiliza la configuración Aplicada al propietario del bucket en S3 Object Ownership. La configuración Aplicada al propietario del bucket no admite concesiones de destino. Para obtener más información, consulte Permisos para entrega de registros.

Solución de los errores de entrega

Para evitar problemas con el registro de acceso al servidor, siga estas prácticas recomendadas:

- El grupo de entrega de registros de S3 tiene acceso de escritura al bucket de destino: el grupo de entrega de registros de S3 entrega los registros de acceso al servidor al bucket de destino. Se puede utilizar una política de bucket o una lista de control de acceso (ACL) para otorgar acceso de escritura al bucket de destino. Sin embargo, le recomendamos que utilice una política de bucket en lugar de una ACL. Para obtener más información acerca de cómo otorgar acceso de escritura al bucket de destino, consulte Permisos para entrega de registros.

  > **Note**
  >
  > Si el bucket de destino utiliza la configuración Aplicada al propietario del bucket para Propiedad del objeto, tenga en cuenta lo siguiente:
  >
  > - Las ACL están desactivadas y ya no afectan a los permisos. Esto significa que no puede actualizar la ACL del bucket para conceder acceso al grupo de entrega de registros de S3. En su lugar, debe actualizar la política del bucket de destino para conceder acceso a la entidad principal del servicio de registro.
  > - Tampoco puede incluir concesiones de destino en la configuración de `PutBucketLogging`.

- La política de bucket para el bucket de destino permite el acceso a los registros: compruebe la política de bucket para el bucket de destino. En la política de bucket, busque cualquier instrucción que contenga `"Effect": "Deny"`. A continuación, compruebe que la instrucción `Deny` no impida que los registros de acceso se escriban en el bucket.
- El bloqueo de objetos de S3 no está activado en el bucket de destino: compruebe si el bucket de destino tiene activado el bloqueo de objetos. El bloqueo de objetos bloquea la entrega del registro de acceso al servidor. Debe elegir un bucket de destino que no tenga activado el bloqueo de objetos.
- Se seleccionan claves administradas de Amazon S3 (SSE-S3) si el cifrado predeterminado está habilitado en el bucket de destino: puede usar el cifrado del bucket predeterminado en el bucket de destino solo si utiliza el cifrado del lado del servidor con claves administradas de Amazon S3

(SSE-S3). El cifrado del lado del servidor con claves de AWS Key Management Service (AWS KMS) (SSE-KMS) no se admite para los buckets de destino de registro de acceso al servidor. Para obtener más información acerca de cómo habilitar el cifrado predeterminado, consulte Configuración del cifrado predeterminado.

- El bucket de destino no tiene la opción Pago por solicitante habilitada: no se admite el uso de un bucket de Pago por solicitante como bucket de destino para el registro de acceso al servidor. Para permitir la entrega de los registros de acceso al servidor, desactive la opción Pago por solicitante en el bucket de destino.
- Revise su política de control de servicios de AWS Organizations: cuando use AWS Organizations, compruebe las políticas de control de servicios para asegurarse de que se permite el acceso a Amazon S3. Las políticas de control de servicios especifican el número máximo de permisos que pueden tener las cuentas afectadas. En la política de control de servicios, busque cualquier instrucción que contenga `"Effect": "Deny"` y verifique que las instrucciones `Deny` no impidan que se escriban registros de acceso en el bucket. Para obtener más información, consulte Políticas de control de servicios (SCP) en la Guía del usuario de AWS Organizations.
- Espere un tiempo hasta que surtan efecto los cambios recientes en la configuración del registro: habilitar el registro de acceso al servidor por primera vez o cambiar el bucket de destino para los registros requieren un tiempo hasta que surtan efecto. Es posible que todas las solicitudes tarden más de una hora en registrarse y entregarse correctamente.

  Para comprobar los fallos de entrega de registros, active las métricas de solicitudes en Amazon CloudWatch. Si los registros no se entregan en unas pocas horas, busque la métrica `4xxErrors`, que puede indicar errores en la entrega de registros. Para obtener más información acerca de la activación de las métricas de solicitudes, consulte the section called “Creación de una configuración de métricas para todos los objetos”.

Solucionar problemas de control de versiones

Los siguientes temas pueden ayudarlo a solucionar problemas habituales de control de versiones de Amazon S3.

Temas

- Quiero recuperar objetos que se han eliminado por error en un bucket con el control de versiones activado
- Quiero eliminar los objetos versionados de forma permanente

- Estoy experimentando una bajada del rendimiento después de habilitar el control de versiones de buckets

Quiero recuperar objetos que se han eliminado por error en un bucket con el control de versiones activado

En general, cuando se eliminan versiones de objetos de los buckets de S3, Amazon S3 no puede recuperarlas. Sin embargo, si has activado el control de versiones de S3 en el bucket de S3, una solicitud `DELETE` que no especifique un ID de versión no puede eliminar un objeto de forma permanente. En su lugar, se añade un marcador de eliminación como marcador de posición. El marcador de eliminación se convierte en la versión actual del objeto.

Para comprobar si los objetos eliminados se eliminan de forma permanente o temporal (con un marcador de eliminación en su lugar), haga lo siguiente:

1. Inicie sesión AWS Management Console Management Console y abra la consola de Amazon S3 en https://console.aws.amazon.com/s3/.
2. En el panel de navegación izquierdo, elija Instancias.
3. En la lista Buckets (Buckets), elija el nombre del bucket que contiene el objeto.
4. En la lista Objetos, active la opción Mostrar versiones a la derecha de la barra de búsqueda y, a continuación, busque el objeto eliminado en la barra de búsqueda. Esta opción solo está disponible si el control de versiones ya estaba activado anteriormente en el bucket.

   También puede usar Inventario de S3 para buscar objetos eliminados.
5. Si no encuentra el objeto después de seleccionar Mostrar versiones o crear un informe de inventario, y tampoco encuentra un marcador de eliminación del objeto, la eliminación será permanente y no podrá recuperar el objeto.

También puede verificar el estado de un objeto eliminado mediante la operación de la API `HeadObject` de la AWS Command Line Interface (AWS CLI). Para ello, el comando `head-object` siguiente y sustituya *`user input placeholders`* con su información:

```
aws s3api head-object --bucket amzn-s3-demo-bucket --key index.html
```

Si ejecuta el comando `head-object` en un objeto versionado cuya versión actual es un marcador de eliminación, aparecerá un error 404 No encontrado. Por ejemplo:

Se ha producido un error (404) al llamar a la operación HeadObject: No encontrada

Si ejecuta el comando `head-object` en un objeto versionado y proporciona el ID de versión del objeto, Amazon S3 recupera los metadatos del objeto y confirma que el objeto sigue existiendo y no se ha eliminado de forma permanente.

```
aws s3api head-object --bucket amzn-s3-demo-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 encuentra el objeto y la versión más reciente es un marcador de eliminación, la versión anterior del objeto seguirá existiendo. Como el marcador de eliminación es la versión actual del objeto, puede recuperarlo borrando el marcador de eliminación.

Tras eliminar permanentemente el marcador de eliminación, la segunda versión más reciente del objeto pasa a ser la versión actual del objeto, lo que hace que el objeto vuelva a estar disponible. Para obtener una representación visual de cómo se recuperan los objetos, consulte Borrar marcadores de eliminación.

Para quitar una versión específica de un objeto, debe ser el propietario del bucket. Para eliminar permanentemente un marcador de eliminación, se debe incluir el ID de versión en una solicitud `DeleteObject`. Para borrar el marcador de eliminación, use el siguiente comando y sustituya *`user input placeholders`* con su información:

```
aws s3api delete-object --bucket amzn-s3-demo-bucket --key index.html --
version-id versionID
```

Para obtener más información sobre el comando `delete-object`, consulte delete-object en la Referencia de comandos de la AWS CLI. Para obtener más información acerca de cómo eliminar marcadores de eliminación de forma permanente, consulte Gestión de marcadores de eliminación.

Quiero eliminar los objetos versionados de forma permanente

En un bucket con control de versiones, una solicitud `DELETE` sin un ID de versión no puede eliminar un objeto de forma permanente. En cambio, dicha solicitud inserta un marcador de eliminación.

Para eliminar objetos versionados de forma permanente, puede elegir entre los siguientes métodos:

- Crear una regla de ciclo de vida de S3 para eliminar las versiones no actuales de forma permanente. Para eliminar versiones anteriores de objetos de manera permanente, seleccione Eliminar definitivamente versiones de objetos no actuales y, a continuación, escriba un número en Días tras los que los objetos dejan de ser actuales. Puede especificar opcionalmente el número de versiones más recientes que desea retener introduciendo un valor en Number of newer versions to retain (Número de versiones más recientes que se deben retener). Para obtener más información sobre la creación de esta regla, consulte Configurar un ciclo de vida de S3.
- Eliminar una versión especificada al incluir el ID de la versión en la solicitud `DELETE`. Para obtener más información, consulte Eliminar versiones de objetos definitivamente.
- Crear una regla de ciclo de vida para que caduquen las versiones actuales. Para que caduquen las versiones actuales de los objetos, seleccione Hacer que venzan las versiones actuales de los objetos y escriba un número en Días después de la creación del objeto. Para obtener más información sobre la creación de esta regla de ciclo de vida, consulte Configurar un ciclo de vida de S3.
- Para eliminar definitivamente todos los objetos versionados y los marcadores de eliminación, cree dos reglas de ciclo de vida: una para que venzan las versiones actuales y eliminar las versiones no actuales de los objetos definitivamente y otra para borrar los marcadores de eliminación de los objetos caducados.

En un bucket con control de versiones activado, una solicitud `DELETE` que no especifique un ID de versión solo puede eliminar objetos con un ID de versión `NULL`. Si el objeto se cargó cuando se activó el control de versiones, una solicitud `DELETE` que no especifique un ID de versión creará un marcador de eliminación de ese objeto.

> **Note**
>
> En el caso de los buckets con bloqueo de objetos de S3 activado, una solicitud de objeto `DELETE` con un ID de versión de objeto protegido genera un error 403 Acceso denegado. Una solicitud de objeto `DELETE` sin un ID de versión añade un marcador de eliminación como la versión más reciente del objeto con una respuesta 200 OK. Los objetos protegidos

por bloqueo de objetos no se pueden eliminar definitivamente hasta que se eliminen sus periodos de retención y retenciones legales correspondientes. Para obtener más información, consulte the section called “Cómo funciona Bloqueo de objetos de S3”.

Estoy experimentando una bajada del rendimiento después de habilitar el control de versiones de buckets

Si hay demasiados marcadores de eliminación u objetos versionados y si no se siguen las prácticas recomendadas, se puede producir una bajada del rendimiento en los buckets habilitados para el control de versiones.

Demasiados marcadores de eliminación

Al activar el control de versiones en un bucket, una solicitud `DELETE` sin ID de versión realizada a un objeto crea un marcador de eliminación con un ID de versión exclusivo. Las configuraciones del ciclo de vida con la regla Hacer que venzan las versiones actuales de los objetos añaden un marcador de eliminación con un ID de versión único a cada objeto. El exceso de marcadores de eliminación puede reducir el rendimiento del bucket.

Cuando se suspende control de versiones en un bucket, Amazon S3 marca el ID de versión como `NULL` en los objetos que se acaban de crear. Una acción de vencimiento en un bucket con el control de versiones suspendido provoca que Amazon S3 cree un marcador de eliminación con un ID de versión `NULL`. En un bucket con control de versiones suspendido, se crea un marcador de eliminación `NULL` para cualquier solicitud de eliminación. Estos marcadores de eliminación `NULL` también se denominan marcadores de eliminación objetos vencidos cuando se eliminan todas las versiones de objetos y solo queda un único marcador de eliminación. Si se acumulan demasiados marcadores de eliminación `NULL`, el rendimiento en el bucket se ve afectado.

Demasiados objetos versionados

Si un bucket con control de versiones habilitado contiene objetos con millones de versiones, puede producirse un aumento de los errores 503 Servicio no disponible. Si detecta un aumento significativo en el número de respuestas de HTTP 503 Servicio no disponible recibidas para solicitudes de objeto `PUT` o `DELETE` en un bucket de Amazon S3 con el control de versiones habilitado, puede que tenga uno o varios objetos en el bucket con millones de versiones. Si tiene objetos con millones de versiones, Amazon S3 limita automáticamente las solicitudes al bucket. Las solicitudes de limitación protegen al bucket de una cantidad excesiva de tráfico de solicitudes, lo que podría impedir que se realicen otras solicitudes al mismo bucket.

Para determinar qué objetos tienen millones de versiones, utilice el inventario de S3. El inventario de S3 genera un informe que crea una lista de archivos sin formato de los objetos de un bucket. Para obtener más información, consulte Inventario de Amazon S3.

Para comprobar si hay un número elevado de objetos versionados en el bucket, utilice las métricas de Lente de almacenamiento de S3 para ver el Recuento de objetos de la versión actual, Recuento de objetos de la versión no actual de Recuento de objetos de marcador de eliminación. Para obtener más información acerca de las métricas de la lente de almacenamiento, consulte Glosario de métricas de Amazon S3 Storage Lens.

El equipo de Amazon S3 insta a los clientes a investigar las aplicaciones que sobrescriben repetidamente el mismo objeto y pueden llegar a generar millones de versiones de ese objeto para determinar si la aplicación funciona según lo previsto. Por ejemplo, una aplicación que sobrescriba el mismo objeto cada minuto durante una semana puede crear más de diez mil versiones. Recomendamos almacenar menos de cien mil versiones para cada objeto. Si tiene un caso de uso que requiere millones de versiones para uno o más objetos, contacte con el equipo de AWS Support para solicitar ayudar para buscar una solución mejor.

Prácticas recomendadas

Se recomienda aplicar las siguientes prácticas recomendadas para evitar problemas de reducción del rendimiento relacionados con el control de versiones:

- Habilite una regla de ciclo de vida para que haga que venzan las versiones anteriores de los objetos. Por ejemplo, puede crear una regla de ciclo de vida para que caduquen las versiones no actuales transcurridos 30 días desde la última actualización del objeto. También puede conservar varias versiones no actuales si no quiere eliminarlas todas. Para obtener más información, consulte Configurar un ciclo de vida de S3.
- Habilite una regla de ciclo de vida para eliminar los marcadores de eliminación de objetos caducados que no tengan objetos de datos asociados en el bucket. Para obtener más información, consulte Eliminar marcadores de eliminación de objetos que vencieron.

Para obtener más información sobre las prácticas recomendadas de optimización del rendimiento de S3, consulte Prácticas recomendadas para patrones de diseño.

Obtención de los ID de las solicitudes de Amazon S3 para AWS Support

Siempre que contacte con AWS Support porque han surgido errores o se ha encontrado con un comportamiento inesperado en Amazon S3, debe proporcionar los ID de las solicitudes relacionadas con la acción que ha fallado. AWS Support utiliza estos ID de las solicitudes para ayudarle a resolver los problemas que está experimentando.

Las ID de solicitudes vienen en pares, se devuelven en cada respuesta que procesa Amazon S3 (incluso las erróneas) y se pueden obtener a través de registros detallados. Hay una serie de métodos comunes para obtener los ID de solicitud, incluidos los registros de acceso de S3 y los eventos o eventos de datos de AWS CloudTrail.

Después de recuperar estos registros, copie y conserve esos dos valores porque los necesitará cuando contacte con AWS Support. Para obtener más información sobre cómo contactar con AWS Support, consulte Contacte con AWS o la Documentación de AWS Support.

Utilización de HTTP para obtener los ID de las solicitudes

Para obtener sus ID de solicitudes, `x-amz-request-id` y `x-amz-id-2` puede registrar los detalles de una solicitud HTTP antes de que llegue a la aplicación de destino. Existen diversas herramientas de terceros que se pueden utilizar para recuperar registros detallados para solicitudes HTTP. Elija una de confianza y ejecútela para escuchar en el puerto por el que circula su tráfico de Amazon S3 al mismo tiempo que envía otra solicitud HTTP de Amazon S3.

Para solicitudes HTTP, el par de ID de las solicitudes se verá de la siguiente manera:

```
x-amz-request-id: 79104EXAMPLEB723
x-amz-id-2: IOWQ4fDEXAMPLEQM+ey7N9WgVhSnQ6JEXAMPLEZb7hSQDASK+Jd1vEXAMPLEa3Km
```

> **Note**
>
> Las solicitudes HTTPS se cifran y ocultan en la mayoría de las capturas de paquetes.

Utilización de un navegador web para obtener ID de solicitudes

La mayoría de los navegadores web tienen herramientas para desarrolladores que le permiten ver encabezados de solicitudes.

Para las solicitudes basadas en navegador web que devuelven un error, el par de ID de solicitudes se verá como en los siguientes ejemplos.

```
<Error><Code>AccessDenied</Code><Message>Access Denied</Message>
<RequestId>79104EXAMPLEB723</RequestId><HostId>IOWQ4fDEXAMPLEQM
+ey7N9WgVhSnQ6JEXAMPLEZb7hSQDASK+Jd1vEXAMPLEa3Km</HostId></Error>
```

Para obtener el par de ID de solicitudes de solicitudes realizadas correctamente, utilice las herramientas para desarrolladores de su navegador para ver los encabezados de respuesta HTTP. Para obtener información acerca de las herramientas para desarrolladores para navegadores específicos, consulte Solución de problemas de Amazon S3 - Cómo recuperar sus ID de solicitudes de S3 en AWS re:Post.

Uso de los SDK de AWS para obtener los ID de solicitudes

En las siguientes secciones, se incluye información para configurar registros con un SDK de AWS. Si bien puede habilitar registros detallados en cada solicitud y respuesta, no se recomienda habilitar registros en sistemas de producción, porque unas respuestas o solicitudes de gran tamaño pueden causar un retraso significativo en una aplicación.

En el caso de las solicitudes del SDK de AWS, el par de ID de solicitudes se verá como en los siguientes ejemplos.

```
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
```

Uso del SDK para Go para obtener ID de solicitudes

Puede configurar el registro mediante SDK para Go. Para obtener más información, consulte Metadatos de respuesta en la Guía para desarrolladores de SDK para Go V2.

Uso del SDK para PHP para obtener ID de solicitudes

Puede usar PHP para configurar el registro. Para obtener más información, consulte ¿Cómo puedo ver qué datos se envían a través de la red? en la Guía para desarrolladores de AWS SDK for PHP.

Uso del SDK para Java para obtener ID de solicitudes

Puede habilitar el registro para solicitudes o respuestas específicas para detectar y devolver solo los encabezados relevantes. Para ello, importe la clase `com.amazonaws.services.s3.S3ResponseMetadata`. Después, podrá guardar la solicitud en una variable antes de ejecutar la solicitud real. Llame a `getCachedResponseMetadata(AmazonWebServiceRequest request).getRequestID()` para obtener la solicitud o respuesta registrada.

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());
```

Además, puede utilizar registros detallados de cada solicitud y respuesta de Java. Para obtener más información, consulte Registro detallado en red en la Guía del desarrollador de AWS SDK for Java.

Uso de AWS SDK for .NET para obtener ID de solicitudes

Puede configurar registros en AWS SDK for .NET con la herramienta de registro `System.Diagnostics` integrada. Para obtener más información, consulte la publicación sobre el registro con AWS SDK for .NET del blog para desarrolladores de AWS.

> **Note**
>
> De forma predeterminada, el registro devuelto solo incluye información de errores. El archivo de configuración debe incluir `AWSLogMetrics` (y, de forma opcional, `AWSResponseLogging`) para obtener los ID de solicitudes.

Uso del SDK para Python (Boto3) para obtener los ID de solicitudes

AWS SDK for Python (Boto3) le permite registrar respuestas específicas. Puede utilizar esta función para capturar solo los encabezados relevantes. En el siguiente código se muestra cómo registrar partes de la respuesta en un archivo:

```
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'])
```

También puede detectar excepciones y registrar información relevante cuando se produce una excepción. Para obtener más información, consulte el temas sobre cómo obtener información útil de las respuestas de error en la Referencia de la API de AWS SDK para Python (Boto).

Además, puede configurar Boto3 para generar registros de depuración detallados mediante el siguiente código:

```
import boto3
boto3.set_stream_logger('', logging.DEBUG)
```

Para obtener más información, consulte set_stream_logger en la Referencia de la API de AWS SDK para Python (Boto).

Uso del SDK para Ruby para obtener los ID de solicitudes

Para obtener sus ID de solicitudes puede usar las versiones 1, 2 o 3 del SDK para Ruby.

- Si utiliza la versión 1 del SDK para Ruby: puede activar el registro en red HTTP a nivel global con la siguiente línea de código.

  ```
  s3 = AWS::S3.new(:logger => Logger.new($stdout), :http_wire_trace => true)
  ```

- Si utiliza la versión 2 o la versión 3 del SDK para Ruby: puede activar el registro en red HTTP a nivel global con la siguiente línea de código.

  ```
  s3 = Aws::S3::Client.new(:logger => Logger.new($stdout), :http_wire_trace => true)
  ```

Para obtener consejos sobre la obtención de información de red de un cliente AWS, consulte Consejos para la depuración: Obtener información del rastro de red de un cliente.

Uso de AWS CLI para obtener ID de solicitudes

Para obtener los ID de solicitudes cuando utilices la AWS Command Line Interface (AWS CLI), agregue `--debug` al comando .

Uso de Windows PowerShell para obtener ID de solicitudes

Para obtener información sobre la recuperación de registros con Windows PowerShell, consulte la entrada del blog sobre el registro de respuestas en AWS Tools for Windows PowerShell .NET Development.

Uso de eventos de datos de AWS CloudTrail para obtener ID de solicitudes

Un bucket de Amazon S3 configurado con los eventos de datos de CloudTrail para registrar operaciones de la API de nivel de objeto de S3 proporciona información detallada sobre las acciones que realiza un usuario, un rol o un servicio de AWS en Amazon S3. Puede identificar los ID de solicitudes de S3 consultando los eventos de CloudTrail con Athena.

Uso del registro de acceso al servidor de S3 para obtener los ID de solicitudes

Un bucket de Amazon S3 configurado para el registro de acceso al servidor S3 proporciona registros detallados para cada solicitud realizada al bucket. Para identificar los ID de solicitudes de S3, consulte los registros de acceso al servidor mediante Athena.

Historial de revisión

- Versión actual de API: 2006-03-01

En la siguiente tabla se describen los cambios importantes en cada versión de la Referencia de la API de Amazon Simple Storage Service y la Guía del usuario de Amazon S3. Para recibir notificaciones sobre los cambios en esta documentación, puede suscribirse a una fuente RSS.

Cambio	Descripción	Fecha
Amazon S3 Select ya no está disponible para los nuevos clientes.	Amazon S3 Select ya no está disponible para los nuevos clientes. Los clientes actuales de Amazon S3 Select pueden seguir utilizando la caracterí stica de la forma habitual. Más información	25 de julio de 2024
Inventario de Amazon S3 admite la clave de condición s3:InventoryAccessibleOptio nalFields	Inventario de Amazon S3 admite la clave de condición s3:InventoryAccessibleOptio nalFields para controlar si los usuarios pueden incluir campos de metadatos opcionales en sus informes. Para obtener más información, consulte Control de la creación de la configuración del informe de inventario de S3.	20 de febrero de 2024
Compatibilidad de IPv6 con S3 en Outposts	Ahora puede acceder a buckets de S3 en Outposts mediante IPv6 a través de los puntos de conexión de doble pila de S3 en Outposts. La compatibilidad de IPv6 con S3	16 de enero de 2024

	en Outposts permite gestionar los buckets de S3 en Outposts y los recursos del plano de control a través de redes IPv6.	
Nueva clase de almacenamiento Amazon S3 en una única zona de alto rendimiento: S3 Express One Zone	Amazon S3 Express One Zone es una clase de almacenamiento de Amazon S3 en zona única de alto rendimiento que está diseñada específicamente para ofrecer acceso constante a los datos en milisegundos de un solo dígito para los datos a los que accede para las aplicaciones sensibles a la latencia. Para obtener más información, consulte S3 Express One Zone.	28 de noviembre de 2023
Mountpoint para Amazon S3 añade compatibilidad con S3 Express One Zone	Ahora puede montar buckets de directorio de S3 Express One Zone con Mountpoint.	28 de noviembre de 2023
Versión del esquema de invocación de Lambda	Operaciones por lotes de Amazon S3 presenta una nueva versión del esquema de invocación de Lambda para usarla con trabajos de Operaciones por lotes que actúan en buckets de directorio. Para obtener más información, consulte Uso de Lambda y Operaciones por lotes de Amazon S3 con buckets de directorio.	28 de noviembre de 2023

Acción de importación para buckets de directorio	Amazon S3 introduce la acción de importación. La importación es un método simplificado para crear trabajos de Operaciones por lotes de Amazon S3 a fin de copiar objetos de buckets de uso general a buckets de directorio. Para obtener más información, consulte Importación de objetos a un bucket de directorio.	28 de noviembre de 2023
Administración del acceso a S3 con S3 Access Grants	Amazon S3 Access Grants le permite administrar los permisos de datos a escala para entidades principal es de AWS Identity and Access Management (IAM), además de las identidades de directorio de los directori os corporativos, como Azure AD. Ahora puede aplicar los permisos de S3 con privilegios mínimos y escalarlos fácilment e según las necesidades de su empresa. Para obtener más información, consulte Administración del acceso con S3 Access Grants.	26 de noviembre de 2023
Mountpoint para Amazon S3 añade la característica de almacenamiento en caché	Con Mountpoint, ahora puede configurar el almacenamiento en caché para los datos a los que se accede repetidamente.	22 de noviembre de 2023

Mejora de la generación de manifiestos de Operaciones por lotes de Amazon S3	Ahora puede indicar a Operaciones por lotes de Amazon S3 que genere un manifiesto automáticamente en función de los criterios de filtro de objetos que especifique al crear su trabajo. Esta opción está disponible para trabajos de replicación por lotes que cree en la consola de Amazon S3 o para cualquier tipo de trabajo que cree mediante la AWS CLI, los SDK de AWS o la API de REST de Amazon S3. Para obtener más información, consulte Creación de un trabajo de operaciones por lotes de Amazon S3.	22 de noviembre de 2023
Los buckets de Amazon S3 existentes ahora pueden añadir configuraciones de Bloqueo de objetos	Ahora puede habilitar Bloqueo de objetos en un bucket de Amazon S3 existente. Puede establecer retenciones legales y periodos de retención para los buckets nuevos o existentes. Para obtener más información, consulte Usar Bloqueo de objetos.	20 de noviembre de 2023

Métricas de solicitud de S3 Lente de almacenamiento para prefijos	S3 Lente de almacenamiento presenta las métricas de solicitud para prefijos dentro de un bucket de Amazon S3. Para obtener más información, consulte Categorías de métricas.	17 de noviembre de 2023
Grupos de Amazon S3 Storage Lens	S3 Lente de almacenamiento presenta los grupos de Lente de almacenamiento, un filtro definido y personalizado para objetos basado en metadatos de objetos. Para obtener más información, consulte Trabajo con grupos de Amazon S3 Storage Lens.	15 de noviembre de 2023
Nueva política de IAM	S3 en Outposts presenta `AWSServiceRoleForS3OnOutposts`, un rol vinculado a un servicio para ayudarle a administrar los recursos de la red. Para obtener más información, consulte Uso de roles vinculados a servicios para S3 en Outposts.	3 de octubre de 2023

Amazon S3 proporciona el tiempo `Last-Modified` para eliminar marcadores	Amazon S3 proporciona los marcadores de tiempo de eliminación `Last-Modified` en los encabezados de respuesta de S3 `Head` y las operaciones de la API `Get`. Para obtener más información, consulte Trabajar con marcadores de eliminación.	27 de septiembre de 2023
Actualización de Amazon S3 a la política administrada por AWS	Amazon S3 agregó permisos `s3:Describe*` a `AmazonS3ReadOnlyAccess`. Para obtener más información, consulte Políticas administradas por AWS para Amazon S3.	11 de agosto de 2023
Tiempos de inicio mejorados para las solicitudes de restauración estándar realizadas a través de Operaciones por lotes de S3	Las recuperaciones estándar para las solicitudes de restauración que se realizan a través de las operaciones por lotes de S3 ahora pueden iniciarse en cuestión de minutos. Para obtener más información, consulte Opciones de recuperación de archivos.	9 de agosto de 2023

Se agregó Mountpoint, un cliente de alto rendimiento para montar un bucket de Amazon S3 como un sistema de archivos local.	Con Mountpoint, sus aplicaciones pueden acceder a objetos almacenados en Amazon S3 a través de operaciones de archivos, lo que proporciona a sus aplicaciones acceso al almacenamiento elástico y al rendimiento de Amazon S3 a través de una interfaz de archivos.	9 de agosto de 2023
Cifrado del servidor de doble capa con claves de AWS Key Management Service (DSSE-KMS)	Al utilizar el cifrado del servidor de doble capa con claves de AWS Key Management Service (AWS KMS) (DSSE-KMS), se aplican dos capas de cifrado a los objetos cuando se cargan en Amazon S3. Para obtener más información, consulte Uso del cifrado del servidor de doble cara con claves AWS KMS.	13 de junio de 2023

Amazon S3 habilita el S3 Block Public Access y desactiva las listas de control de acceso (ACL) de S3 para todos los buckets nuevos.	Amazon S3 ahora habilita el S3 Block Public Access y desactiva las listas de control de acceso (ACL) de S3 para todos los buckets de S3 nuevos en todas las regiones de AWS. Para obtener más información, consulte Bloquear el acceso público a su almacenamiento de Amazon S3 y Control de la propiedad de los objetos y desactivación de las ACL del bucket.	27 de abril de 2023
Métrica para errores de las operaciones de replicación de S3	Amazon S3 agrega una nueva métrica Amazon CloudWatch para monitorear los errores de replicación de S3. Para obtener más información, consulte Monitoreo del progreso con métricas de replicación.	5 de abril de 2023
DNS privado	AWS PrivateLink para Amazon S3 ya es compatible con el DNS privado. Para obtener más información, consulte DNS privado.	14 de marzo de 2023
La consola de Amazon S3 admite los puntos de acceso entre cuentas	Amazon S3 ahora admite la creación de puntos de acceso entre cuentas con la consola de Amazon S3. Para obtener más información, consulte Creación de puntos de acceso.	14 de marzo de 2023

Amazon S3 en Outposts admite Replicación de S3 en Outposts	Con la replicación de S3 local, puede replicar automáticamente objetos en un solo bucket de Outposts de destino o en varios buckets de destino. Los buckets de destino pueden estar en diferentes AWS Outposts o dentro del mismo Outposts que el bucket de origen. Para obtener más información, consulte Replicación de objetos para S3 en Outposts.	14 de marzo de 2023
Alias de puntos de acceso de Amazon S3 Object Lambda	Al crear un punto de acceso de Object Lambda, Amazon S3 genera automáticamente un alias único para el punto de acceso de Object Lambda. Puede utilizar este alias en lugar de un nombre de bucket de Amazon S3 o el nombre de recurso de Amazon (ARN) del punto de acceso de Object Lambda en las operaciones de plano de datos de punto de acceso. Para obtener más información, consulte Cómo usar un alias de tipo bucket para el punto de acceso de Object Lambda.	14 de marzo de 2023

Amazon S3 admite puntos de acceso de varias regiones entre cuentas	Amazon S3 ahora admite la creación de puntos de acceso de varias regiones entre cuentas con la consola de Amazon S3. Para obtener más información, consulte Creación de puntos de acceso de varias regiones.	14 de marzo de 2023
Puntos de acceso entre cuentas	Amazon S3 admite la creación de puntos de acceso entre cuentas. Puede crear un punto de acceso entre cuentas mediante AWS Command Line Interface (AWS CLI) o la operación `CreateAccessPoint` de API de REST. Para obtener más información, consulte Creación de puntos de acceso.	30 de noviembre de 2022

Amazon S3 admite controles de conmutación por error para puntos de acceso de varias regiones de Amazon S3	Amazon S3 incorpora controles de conmutación por error para puntos de acceso de varias regiones. Estos controles le permiten cambiar el tráfico de solicitudes de acceso a datos de S3 que se envía a través de un punto de acceso multirregional de Amazon S3 a una Región de AWS alternativa en cuestión de minutos para probar y crear aplicaciones de alta disponibilidad. Para obtener más información, consulte Controles de conmutación por error de punto de acceso multirregional de Amazon S3.	28 de noviembre de 2022
La lente de almacenamiento de Amazon S3 aumenta la visibilidad de toda la organización con 34 métricas nuevas	Lente de almacenamiento de S3 incorpora 34 métricas adicionales para descubrir oportunidades de optimización de costos más profundas, identificar prácticas recomendadas de protección de los datos y mejorar el rendimiento de las cargas de trabajo de las aplicaciones. Para obtener más información, consulte Métricas de Lente de almacenamiento de S3.	17 de noviembre de 2022

Amazon S3 admite velocidades de solicitud de restauración más altas para S3 Glacier Flexible Retrieval y S3 Glacier Deep Archive	Amazon S3 admite solicitudes de restauración a una velocidad de hasta 1000 transacciones por segundo, según la Cuenta de AWS para clases de almacenamiento de S3 Glacier Flexible Retrieval y S3 Glacier Deep Archive.	15 de noviembre de 2022
Amazon S3 en Outposts admite acciones y filtros del ciclo de vida de S3 adicionales	S3 en Outposts admite reglas adicionales del ciclo de vida de S3 para optimizar la administración de la capacidad. Puede hacer vencer los objetos a medida que envejecen o se sustituyan por versiones más recientes. Puede crear una regla del ciclo de vida para todo un bucket o un subconjunto de objetos en un bucket filtrando por prefijos, etiquetas de objetos o tamaño de objetos. Para obtener más información, consulte Creación y administración de una configuración del ciclo de vida.	2 de noviembre de 2022

Soporte de Replicación de S3 para objetos SSE-C	Puede replicar los objetos creados con cifrado del lado del servidor con claves proporcionadas por los clientes. Para obtener más información sobre la replicación de objetos cifrados, consulte Replicación de objetos creados con cifrado del servidor (SSE-C, SSE-S3, SSE-KMS).	24 de octubre de 2022
Amazon S3 en Outposts admite alias de puntos de acceso	Con S3 en Outposts, debe utilizar puntos de acceso para acceder a cualquier objeto de un bucket de Outposts. Cada vez que se crea un punto de acceso para un bucket, S3 en Outposts genera de forma automática un alias de punto de acceso. Puede utilizar este alias de punto de acceso en lugar de un ARN de punto de acceso para cualquier operación del plano de datos. Para obtener más información, consulte Uso de un alias de estilo de bucket para su punto de acceso de bucket de S3 en Outposts.	21 de octubre de 2022

S3 Object Lambda admite las operaciones `HeadObject`, `ListObjects` y `ListObjectsV2`	Puede utilizar código personalizado para modificar los datos que devuelven solicitudes `GET`, `LIST` o `HEAD` de S3 estándar para filtrar columnas, redimensionar imágenes de forma dinámica, ocultar datos confidenciales y más. Para obtener más información, consulte Transformación de objetos con S3 Object Lambda.	4 de octubre de 2022
Amazon S3 en Outposts admite el control de versiones de S3	Cuando está habilitado, el control de versiones de S3 guarda diversas copias de un objeto en el mismo bucket. Puede utilizar el control de versiones de S3 para conservar, recuperar y restaurar todas las versiones de los objetos almacenados en su bucket de Outposts. EL control de versiones de S3 ayuda a recuperarse de acciones no deseadas del usuario y de errores de la aplicación. Para obtener más información, consulte Managing S3 Versioning for your S3 on Outposts bucket (Administración de S3 Versioning para su bucket de S3 en Outposts).	21 de septiembre de 2022

AWS Backup para Amazon S3	AWS Backup es un servicio completamente administrado basado en políticas que puede utilizar para definir una política de copia de seguridad central para proteger los datos de Amazon S3. Para obtener más información, consulte Uso de AWS Backup de Amazon S3.	18 de febrero de 2022
Uso de la replicación por lotes de S3 para replicar objetos existentes	Con la replicación por lotes de S3, puede replicar objetos que existían antes de que se estableciera una configuración de replicación. La replicación de los objetos existentes se realiza mediante un trabajo de operaciones por lotes. La replicación por lotes de S3 difiere de la replicación en directo que copia objetos nuevos de forma continua y automática en buckets de Amazon S3. Para obtener más información, consulte Replicación de objetos existentes con la replicación por lotes de S3.	8 de febrero de 2022
Cambio de nombre de S3 Glacier Flexible Retrieval	Se ha cambiado el nombre de la clase de almacenamiento Glacier a S3 Glacier Flexible Retrieval. Este cambio no afecta a la API.	30 de noviembre de 2021

Nueva configuración de S3 Object Ownership para desactivar las ACL	Puede aplicar la configuración de propietario del bucket obligatorio de Object Ownership para desactivar las ACL del bucket y los objetos que contiene y tomar posesión de todos los objetos del bucket. La configuración impuesta por el propietario del bucket simplifica la administración del acceso a los datos almacenados en Amazon S3. Para obtener más información, consulte Control de la propiedad de los objetos y desactivación de las ACL del bucket.	30 de noviembre de 2021
Nueva clase de almacenamiento S3 Intelligent-Tiering	S3 Intelligent-Tiering Archive Instant Access es una clase de almacenamiento adicional en S3 Intelligent-Tiering. Para obtener más información, consulte Cómo funciona S3 Intelligent-Tiering.	30 de noviembre de 2021
Nueva clase de almacenamiento S3 Glacier Instant Retrieval	Ahora puede colocar objetos en la clase de almacenamiento S3 Glacier Instant Retrieval. Para obtener más información acerca de esta clase de almacenamiento, consulte Uso de clases de almacenamiento de Amazon S3.	30 de noviembre de 2021

AWS Backup para la vista previa de Amazon S3	AWS Backup es un servicio completamente administrado basado en políticas que puede utilizar para definir una política de copia de seguridad central para proteger los datos de Amazon S3. Para obtener más información, consulte Uso de AWS Backup de Amazon S3.	30 de noviembre de 2021
AWS Identity and Access Management Access Analyzer para Amazon S3	IAM Access Analyzer ejecuta verificaciones de política para validarla contra la gramática de la política de IAM y las prácticas recomendadas. Para obtener más informaci ón sobre la validación de políticas mediante IAM Access Analyzer, consulte Validació n de políticas de IAM Access Analyzer en la Guía del usuario de IAM.	30 de noviembre de 2021
Nuevos tipos de eventos	Para ver los nuevos tipos de eventos agregados a las notificaciones de eventos de Amazon S3, consulte Notificac iones de eventos de Amazon S3.	29 de noviembre de 2021
Habilitación de Amazon EventBridge en buckets	Para habilitar EventBridge en buckets de Amazon S3 y enviar eventos a Amazon EventBridge, consulte Uso de EventBridge.	29 de noviembre de 2021

Nuevos filtros de S3 Lifecycle	Puede crear reglas del ciclo de vida en función del tamaño del objeto o especificar cuántas versiones de objeto no actuales desea conservar . Para obtener más informaci ón, consulte Ejemplos de configuración del ciclo de vida de S3.	23 de noviembre de 2021
Publicación de métricas de Amazon S3 Storage Lens en Amazon CloudWatch	Puede publicar métricas de actividad y uso de S3 Storage Lens en Amazon CloudWatch para crear una vista unificada del estado operativo en los paneles de CloudWatch. También puede utilizar las características de CloudWatc h, como alarmas y acciones desencadenadas, cálculos de métricas y detección de anomalías para monitorear y tomar medidas en las métricas de S3 Storage Lens. Además, las API de CloudWatch permiten que las aplicacio nes, incluidos proveedores de terceros, accedan a las métricas de S3 Storage Lens. Para obtener más informaci ón, consulte Monitoreo de métricas de S3 Storage Lens en CloudWatch.	22 de noviembre de 2021

Puntos de acceso de varias regiones	Puede utilizar puntos de acceso de varias regiones para crear un punto de conexión global que las aplicaciones puedan utilizar para gestionar solicitudes de buckets de Amazon S3 ubicadas en varias Regiones de AWS. Puede utilizar este punto de acceso de varias regiones para dirigir los datos a un bucket con la latencia más baja. Para obtener más información acerca de los puntos de acceso de varias regiones y cómo utilizarlos, consulte Puntos de acceso de varias regiones de Amazon S3.	2 de septiembre de 2021

Amazon S3 en Outposts agrega acceso local directo a las aplicaciones	Ejecute sus aplicaciones fuera de la Virtual Private Cloud (VPC) de AWS Outposts y acceda a sus datos de S3 en Outposts. También puede acceder a objetos de S3 en Outposts directamente desde la red en las instalaciones. Para obtener más informaci ón acerca de la configura ción de S3 en Outposts con direcciones IP propiedad del cliente (CoIP) y acceder a sus objetos mediante la creación de una gateway localdesde la reden las instalaciones, consulte Acceso a Amazon S3 en Outposts mediante puntos de acceso solo de la VPC.	29 de julio de 2021
Alias de punto de acceso de Amazon S3	Al crear un punto de acceso, Amazon S3 genera de forma automática un alias que usted puede utilizar en lugar de un nombre de bucket para el acceso a datos. Puede utilizar este alias de punto de acceso en lugar de un nombre de recurso de Amazon (ARN) para cualquier operación de plano de datos de punto de acceso. Para obtener más información, consulte Uso de un alias de estilo de bucket para su punto de acceso.	26 de julio de 2021

Inventario de Amazon S3 y Operaciones por lotes de S3 admiten el estado de Clave de bucket de S3	La herramienta de operaciones por lotes y el inventario de Amazon S3 admiten la identificación y la copia de objetos existentes con claves de bucket de S3. Las claves de bucket de S3 aceleran la reducción de los costos de cifrado del lado del servidor para los objetos existentes. Para obtener más información, consulte Inventario de Amazon S3 y Copia de objetos mediante la herramienta de operaciones por lotes.	3 de junio de 2021
Instantánea de la cuenta de métricas de Amazon S3 Storage Lens	La instantánea de la cuenta de S3 Storage Lens muestra el almacenamiento total, el recuento de objetos y el tamaño promedio de los objetos en la página de inicio de la consola de S3 (Buckets) mediante el resumen de las métricas del panel predeterminado. Para obtener más información, consulte la instantánea de la cuenta de métricas de S3 Storage Lens.	5 de mayo de 2021
Aumento de la compatibilidad con los puntos de enlace de Amazon S3 en Outposts	S3 en Outposts ahora admite hasta 100 puntos de enlace por Outpost. Para obtener más información, consulte Restricciones de red S3 en Outposts.	29 de abril de 2021

Notificaciones de eventos de Amazon S3 en Outposts en Amazon CloudWatch Events	Puede utilizar CloudWatch Events para crear una regla para capturar cualquier evento de API de S3 en Outposts y recibir notificaciones a través de todos los destinos compatibles de CloudWatch. Para obtener más informaci ón, consulte Recepción de notificaciones de eventos de S3 en Outposts mediante CloudWatch Events.	19 de abril de 2021
S3 Object Lambda	Con S3 Object Lambda puede agregar su propio código a las solicitudes GET de Amazon S3 para modificar y procesar los datos a medida que vuelven a una aplicación. Puede utilizar código personalizado para modificar los datos que devuelven solicitudes GET de S3 estándar para filtrar columnas, redimensionar imágenes de forma dinámica, ocultar datos confidenc iales y más. Para obtener más información, consulte Transformación de objetos.	18 de marzo de 2021

AWS PrivateLink	Con AWS PrivateLink para Amazon S3, puede conectarse de forma directa a S3 mediante un punto de conexión de interfaz en su nube privada virtual (VPC) en lugar de conectarse a través de Internet. Se puede acceder de manera directa a los puntos de enlace de la interfaz desde aplicaciones que se encuentran en las instalaciones o en una Región de AWS diferente. Para obtener más información, consulte AWS PrivateLink para Amazon S3.	2 de febrero de 2021
Administración de la capacidad de Amazon S3 en Outposts con AWS CloudTrail	Los eventos de administración de S3 en Outposts están disponibles a través de registros de CloudTrail. Para obtener más información, consulte Gestión de la capacidad de S3 en Outposts con CloudTrail.	21 de diciembre de 2020

Consistencia sólida	Amazon S3 proporciona una sólida coherencia de lectura tras escritura para las operaciones PUT y DELETE de objetos del bucket de S3 en todas las Regiones de AWS. Además, las operaciones de lectura en Amazon S3 Select, las listas de control de acceso de Amazon S3, las etiquetas de objetos de Amazon S3 y los metadatos de objetos (por ejemplo, el objeto HEAD) son muy consistentes. Para obtener más información, consulte el Modelo de consistencia de datos de Amazon S3.	1 de diciembre de 2020

Sincronización de modificación de réplica de Amazon S3	La sincronización de modificación de réplica de Amazon S3 mantiene los metadatos de objetos, como etiquetas, listas de control de acceso (ACL) y configuraciones de bloqueo de objetos, sincronizados entre los objetos de origen y las réplicas. Cuando esta función está habilitada, Amazon S3 replica los cambios de metadatos realizados en el objeto de origen o en las copias de réplica. Para obtener más información, consulte Replica de cambios de metadatos con sincronización de modificación de réplica.	1 de diciembre de 2020

Claves de bucket de Amazon S3	Las claves de bucket de Amazon S3 reducen el costo del cifrado del lado del servidor de Amazon S3 con AWS Key Managemen t Service (SSE-KMS). Esta nueva clave de bucket para el cifrado del lado del servidor puede reducir los costos de solicitudes de AWS KMS hasta en un 99 % al disminuir el tráfico de solicitudes de Amazon S3 a AWS KMS. Para obtener más información, consulte Reducción del costo de SSE-KMS utilizando las claves de bucket de S3.	1 de diciembre de 2020

Amazon S3 Storage Lens	Lente de almacenamiento de S3 agrega las métricas y muestra la información en la sección Account snapshot (Instantánea de la cuenta) en la página Buckets de la consola de Amazon S3. S3 Storage Lens también proporciona un panel interactivo que puede utilizar para visualizar información y tendencias, marcar valores atípicos y recibir recomendaciones para optimizar los costes de almacenamiento y aplicar las prácticas recomendadas de protección de datos. El panel tiene opciones de desglose para generar y visualizar información en el nivel de la organización, la cuenta, la Región de AWS, la clase de almacenamiento, el bucket, el prefijo o el grupo de Lente de almacenamiento. También puede enviar una exportación de métricas diaria en formato CSV o Parquet a un bucket de S3. Para obtener más información, consulte Evaluación de la actividad y el uso de almacenamiento con S3 Storage Lens.	18 de noviembre de 2020

Seguimiento de solicitudes de S3 con AWS X-Ray	Amazon S3 se integra con X-Ray para propagar el contexto de seguimiento y ofrecer una cadena de solicitudes con nodos ascendentes y descendentes. Para obtener más información, consulte Rastreo de solicitudes con X-Ray.	16 de noviembre de 2020
Métricas de replicación de S3	Las métricas de replicación de S3 proporcionan métricas detalladas para las reglas de replicación en la configuración de la misma. Para obtener más información, consulte Métricas de replicación y notificaciones de eventos de Amazon S3.	9 de noviembre de 2020
Acceso a archivo y acceso a archivo profundo de S3 Intelligent-Tiering	Acceso a archivo y acceso a archivo profundo de S3 Intelligent-Tiering son niveles de almacenamiento adicionales en S3 Intelligent-Tiering. Para obtener más información, consulte Clase de almacenamiento para optimizar automáticamente los objetos a los que se obtiene acceso de forma frecuente e infrecuente.	9 de noviembre de 2020

Replicación de marcador de eliminación	Con la replicación de marcador de eliminación, puede asegurarse de que los marcadores de eliminación se copian en los buckets de destino para las reglas de replicación. Para obtener más información, consulte Uso de la replicación de marcador de eliminación.	9 de noviembre de 2020
S3 Object Ownership	Propiedad de objetos es una configuración de bucket de S3 que puede usar para controlar la propiedad de nuevos objetos que se cargan en los buckets. Para obtener más información, consulte Uso de S3 Object Ownership.	2 de octubre de 2020

Amazon S3 en Outposts	Con Amazon S3 en Outposts, puede crear buckets de S3 en recursos de AWS Outposts y almacenar y recuperar fácilmente objetos en las instalaciones para las aplicaciones que requieren acceso local a los datos, procesamiento local de los datos y residencia de los datos. Puede utilizar S3 en Outposts a través de la AWS Management Console, la AWS CLI, los SDK de AWS o la API de REST. Para obtener más información, consulte Uso de Amazon S3 en Outposts.	30 de septiembre de 2020
Condición de propietario del bucket	Puede utilizar la condición de propietario del bucket de Amazon S3 para asegurars e de que los buckets que utiliza en las operaciones de S3 pertenecen a las Cuentas de AWS que espera. Para obtener más informaci ón, consulte Condición de propietario del bucket.	11 de septiembre de 2020

Compatibilidad de Operaciones por lotes de S3 para la retención de Bloqueo de objetos	Ahora puede utilizar Operaciones por lotes de S3 con Bloqueo de objetos de S3 para aplicar la configuración de retención a muchos objetos de Amazon S3 al mismo tiempo. Para obtener más información, consulte Establecimiento de fechas de retención de bloqueo de objetos de S3 con Operaciones por lotes de S3.	4 de mayo de 2020
Compatibilidad de Operaciones por lotes de S3 para la retención legal de Bloqueo de objetos	Ahora puede utilizar Operaciones por lotes de S3 con Bloqueo de objetos de S3 para añadir retenciones legales a muchos objetos de Amazon S3 al mismo tiempo. Para obtener más información, consulte Uso de Operaciones por lotes de S3 para establecer la retención legal de Bloqueo de objetos de S3.	4 de mayo de 2020
Etiquetas de trabajo para Operaciones por lotes de S3	Puede añadir etiquetas a los trabajos de Operaciones por lotes de S3 para controlar y etiquetar esos trabajos. Para obtener más información, consulte Etiquetas para trabajos de Operaciones por lotes de S3.	16 de marzo de 2020

Puntos de acceso de Amazon S3	Los puntos de acceso de Amazon S3 simplifican la administración del acceso a los datos a escala para los conjuntos de datos compartidos en S3. Los puntos de acceso son puntos de enlace de red con nombre que están asociados a los buckets que se pueden utilizar para realizar operaciones con objetos de S3. Para obtener más información, consulte Administración del acceso a datos con puntos de acceso de Amazon S3.	2 de diciembre de 2019
Access Analyzer para Amazon S3	Access Analyzer para Amazon S3 avisa que los buckets de S3 están configurados para permitir el acceso a cualquier usuario de Internet u otras Cuentas de AWS, incluidas las cuentas ajenas a la organización. Para obtener más información, consulte Uso de Access Analyzer for Amazon S3.	2 de diciembre de 2019

Control del tiempo de replicación de S3 (S3 RTC)	El control del tiempo de replicación de S3 (S3 RTC) replica la mayoría de los objetos que carga en Amazon S3 en cuestión de segundos, y el 99,99 % de esos objetos en un plazo de 15 minutos. Para obtener más información, consulte Replicación de objetos mediante el control de tiempo de replicación de S3 (S3 RTC).	20 de noviembre de 2019
Replicación en la misma región	Puede utilizar la reproducción en la misma región (SRR) para copiar objetos en buckets de Amazon S3 en la misma Región de AWS. Para obtener información acerca de la replicación entre regiones (CRR) y la replicación en la misma región, consulte Replicación.	18 de septiembre de 2019
Compatibilidad de la replicación entre regiones para el Bloqueo de objetos de S3	La replicación entre regiones ahora admite el bloqueo de objetos. Para obtener más información, consulte ¿Qué replica Amazon S3?.	28 de mayo de 2019

Operaciones por lotes de S3	Con Operaciones por lotes de S3, puede realizar operaciones por lotes a gran escala en objetos de Amazon S3. Operaciones por lotes de S3 puede ejecutar una sola operación en las listas de objetos que especifique. Un solo trabajo puede realizar la operación especificada en miles de millones de objetos que contiene exabytes de datos. Para obtener más información, consulte Realización de operaciones por lotes de S3.	30 de abril de 2019
Región Asia Pacífico (Hong Kong)	Amazon S3 ya está disponible en la región Asia-Pacífico (Hong Kong). Para obtener más información acerca de las regiones y los puntos de conexión de Amazon S3, consulte Regiones y puntos de conexión en Referencia general de AWS.	24 de abril de 2019
Se ha agregado un nuevo campo a los registros de acceso al servidor	Amazon S3 ha agregado el siguiente campo nuevo a los registros de acceso al servidor: versión de Transport Layer Security (TLS). Para obtener más información, consulte Formato de registro de acceso al servidor.	28 de marzo de 2019

Nueva clase de almacenamiento de archivado	Amazon S3 ahora ofrece una nueva clase de almacenamiento de archivado, S3 Glacier Deep Archive (`DEEP_ARCHIVE`), para almacenar objetos a los que se accede con poca frecuencia. Para obtener más información, consulte Clases de almacenamiento.	27 de marzo de 2019
Se han agregado nuevos campos a los registros de acceso al servidor	Amazon S3 ha agregado los siguientes campos nuevos a los registros de acceso al servidor: ID de host, Versión de firma, Conjunto de cifrado, Tipo de autenticación y Encabezado de host. Para obtener más información, consulte Formato de registro de acceso al servidor.	5 de marzo de 2019
Compatibilidad con los archivos de Amazon S3 Inventory con formato Parquet	Amazon S3 admite ahora el formato Apache Parquet (Parquet) además de Apache optimized row columnar (ORC) y el formato de archivo de valores separados con comas (CSV) para los archivos de salida del inventario. Para obtener más información, consulte Inventario.	4 de diciembre de 2018

Bloqueo de objetos de S3	Amazon S3 ofrece ahora la funcionalidad Bloqueo de objetos que proporciona protecciones de escritura única y lectura múltiple para objetos de Amazon S3. Para obtener más información, consulte Bloqueo de objetos.	26 de noviembre de 2018
Restauración de actualización de velocidad	Con la actualización de velocidad de restauración de Amazon S3 puede cambiar a una velocidad de restauración más rápida de la clase de almacenamiento S3 Glacier Flexible Retrieval mientras se está realizando la restauración. Para obtener más información, consulte Restaurar objetos archivados.	26 de noviembre de 2018
Restaurar notificaciones de eventos	Las notificaciones de eventos de Amazon S3 admiten ahora eventos de inicio y finalización al restaurar objetos de la clase de almacenamiento S3 Glacier Flexible Retrieval. Para obtener más información, consulte Notificaciones de eventos.	26 de noviembre de 2018

Operación PUT directamente en la clase de almacenamiento S3 Glacier Flexible Retrieval	La operación PUT de Amazon S3 ahora admite la especificación de S3 Glacier Flexible Retrieval como clase de almacenamiento en el momento en el que crea un objeto. Con anterioridad, había que pasar objetos a la clase de almacenamiento S3 Glacier Flexible Retrieval desde otra clase de almacenamiento de Amazon S3. Además, al usar la replicación entre regiones (CRR) de S3, ahora puede especificar S3 Glacier Flexible Retrieval como clase de almacenamiento para los objetos replicados. Para obtener más información acerca de la clase de almacenamiento S3 Glacier Flexible Retrieval, consulte Clases de almacenamiento. Para obtener más información acerca de la especificación de la clase de almacenamiento para objetos replicados, consulte Información general de la configuración de replicación. Para obtener más información acerca de los cambios directos PUT en la API de REST de S3 Glacier Flexible Retrieval, consulte Historial de revisión: operación	26 de noviembre de 2018

	PUT directamente en S3 Glacier Flexible Retrieval.	
Nueva clase de almacenamiento	Amazon S3 ofrece ahora una clase de almacenamiento nueva llamada S3 Intelligent-Tiering (`INTELLIGENT_TIERING`) diseñada para datos de larga duración, con patrones de acceso desconocidos o cambiantes. Para obtener más información, consulte Clases de almacenamiento.	26 de noviembre de 2018
Amazon S3 Block Public Access	Amazon S3 incluye ahora la capacidad de bloquear el acceso público a buckets y objetos en un bucket o en toda la cuenta. Para obtener más información, consulte Uso de Amazon S3 Block Public Access.	15 de noviembre de 2018

Filtrado de mejoras en reglas de replicación en varias regiones (CRR)	En la configuración de una regla de CRR, puede especific ar un filtro de objeto para elegir un subconjunto de objetos al que aplicar la regla. Anteriormente, solo se podía filtrar en un prefijo de clave de objeto. En esta versión, puede filtrar en un prefijo de clave de objeto, una o varias etiquetas de objeto, o ambos métodos. Para obtener más información, consulte Configuración de la CRR: información general de la configuración de replicación.	19 de septiembre de 2018
Nuevas características de Amazon S3 Select	Amazon S3 Select ahora admite la entrada de Apache Parquet, consultas sobre objetos JSON anidados y dos nuevas métricas de monitoreo de Amazon CloudWatch (`SelectScannedBytes` y `SelectReturnedBytes`).	5 de septiembre de 2018
Actualizaciones ahora disponibles sobre RSS	Ahora puede suscribirse a una fuente RSS para recibir notificaciones sobre actualiza ciones de la Guía del usuario de Amazon S3.	19 de junio de 2018

Actualizaciones anteriores

En la siguiente tabla se describen los cambios importantes de cada versión de la Guía del usuario de Amazon S3 anteriores al 19 de junio de 2018.

Cambio	Descripción	Fecha
Actualización de las muestras de código	Muestras de código actualizadas: • C#: se han actualizado todas las muestras para usar el patrón asíncrono basado en tareas. Para obtener más información, consulte la sección sobre API asíncronas de Amazon Web Services para .NET en la Guía para desarrolladores de AWS SDK for .NET. Ahora las muestras de código son compatibles con la versión 3 del AWS SDK for .NET. • Java: se han actualizado todas las muestras para usar el modelo de compilador de clientes. Para obtener más información sobre el modelo del compilador de clientes, consulte Creación de clientes de servicio. • PHP: se han actualizado todos los ejemplos para utilizar AWS SDK for PHP 3.0. Para obtener más información acerca de AWS SDK for PHP 3.0, consulte AWS SDK for PHP. • Ruby: se ha actualizado el código de muestra para que los ejemplos funcionen con la versión 3 de AWS SDK for Ruby.	30 de abril de 2018
Amazon S3 informa ahora sobre las clases de almacenamiento S3 Glacier Flexible Retrieval y `ONEZONE_IA` para las métricas de almacenamiento de Registros de Amazon CloudWatch	Además de informar sobre los bytes reales, estas métricas de almacenamiento contienen bytes de sobrecarga por objeto para las clases de almacenamiento pertinentes (`ONEZONE_IA` , `STANDARD_IA` y S3 Glacier Flexible Retrieval): • Para los objetos de clase de almacenamiento `ONEZONE_IA` y `STANDARD_IA` , Amazon S3 informa de los objetos de menos de 128 KB como si fueran de	30 de abril de 2018

Cambio	Descripción	Fecha
	128 KB. Para obtener más información, consulte Uso de las clases de almacenamiento de Amazon S3. • Para los objetos de clase de almacenamiento S3 Glacier Flexible Retrieval, las métricas de almacenamiento informan sobre las siguientes sobrecargas: • Una sobrecarga por objeto de 32 KB, cargada en el precio de clase de almacenamiento S3 Glacier Flexible Retrieval • Una sobrecarga por objeto de 8 KB, cargada en el precio de clase de almacenamiento `STANDARD` Para obtener más información, consulte Transición de objetos con Amazon S3 Lifecycle. Para obtener más información acerca de las métricas de almacenamiento, consulte Monitorización de métricas con Amazon CloudWatch.	
Nueva clase de almacenamiento	Amazon S3 ahora ofrece una nueva clase de almacenamiento, `STANDARD_IA` (IA, para acceso poco frecuente) para el almacenamiento de objetos. Esta clase de almacenamiento está optimizada para los datos de duración prolongada y a los que se obtenga acceso con menor frecuencia. Para obtener más información, consulte Uso de las clases de almacenamiento de Amazon S3.	4 de abril de 2018
Amazon S3 Select	Amazon S3 ahora admite la recuperación de contenido de los objetos basándose en una expresión SQL. Para obtener más información, consulte Filtrado y recuperación de datos con Amazon S3 Select.	4 de abril de 2018

Cambio	Descripción	Fecha
Región Asia Pacífico (Osaka-Local)	Amazon S3 ya está disponible en la región Asia-Pacífico (Osaka-Local). Para obtener más información acerca de las regiones y los puntos de conexión de Amazon S3, consulte Regiones y puntos de conexión en la Referencia general de AWS. **Important** Puede utilizar la región Asia-Pacífico (Osaka-Local) únicamente junto con la región Asia-Pacífico (Tokio). Para solicitar acceso a la región Asia-Pacífico (Osaka-Local), póngase en contacto con su representante de ventas.	12 de febrero de 2018
Marca temporal de creación de Amazon S3 Inventory	Amazon S3 Inventory incluye ahora una marca temporal con la fecha y hora de inicio de la creación del informe de Amazon S3 Inventory. Puede utilizar la marca temporal para determinar cambios en su almacenamiento de Amazon S3 desde la hora de inicio cuando se generó el informe de inventario.	16 de enero de 2018
Región de Europa (París)	Amazon S3 ya está disponible en la región UE (París). Para obtener más información acerca de las regiones y los puntos de conexión de Amazon S3, consulte Regiones y puntos de conexión en la Referencia general de AWS.	18 de diciembre de 2017
Región China (Ningxia)	Amazon S3 ya está disponible en la región China (Ningxia) Para obtener más información acerca de las regiones y los puntos de conexión de Amazon S3, consulte Regiones y puntos de conexión en la Referencia general de AWS.	29 de noviembre de 2017

Cambio	Descripción	Fecha
Compatibilidad con los archivos de Amazon S3 Inventory con formato ORC	Amazon S3 admite ahora el formato Apache optimized row columnar (ORC) además del formato de archivo de valores separados con comas (CSV) para los archivos de salida del inventario. Además, ahora puede consultar el inventario de Amazon S3 con el lenguaje SQL estándar mediante Amazon Athena, Amazon Redshift Spectrum y otras herramientas, como Presto, Apache Hive y Apache Spark. Para obtener más información, consulte Inventario de Amazon S3.	17 de noviembre de 2017
Cifrado predeterminado para los buckets de S3	El cifrado predeterminado de Amazon S3 proporciona un medio de definir el comportamiento de cifrado predeterminado para un bucket de S3. Puede configurar el cifrado predeterminado en un bucket para que todos los objetos se cifren cuando se almacenen en el bucket. Los objetos se cifran mediante el cifrado del servidor con claves administradas de Amazon S3 (SSE-S3) o claves administradas por AWS (SSE-KMS). Para obtener más información, consulte Establecer el comportamiento del cifrado predeterminado del lado del servidor para los buckets de Amazon S3.	06 de noviembre de 2017
Estado de cifrado en Amazon S3 Inventory	Amazon S3 permite ahora incluir el estado de cifrado en Amazon S3 Inventory para que pueda saber cómo se cifran los objetos en reposo para los requisitos de conformidad u otros fines. También puede configurar el cifrado de Amazon S3 Inventory con cifrado de lado servidor (SSE) o SSE-KMS, para que todos los archivos del inventario se cifren según corresponda. Para obtener más información, consulte Inventario de Amazon S3.	06 de noviembre de 2017

Cambio	Descripción	Fecha
Mejoras de la replicación entre regiones (CRR)	La replicación entre regiones ahora admite lo siguiente: • En un escenario con varias cuentas, puede agregar la configuración de la CRR para cambiar la propiedad de la réplica a la Cuenta de AWS que posee el bucket de destino. Para obtener más información, consulte Cambiar el propietario de la réplica. • De forma predeterminada, Amazon S3 no reproduce objetos en el bucket de origen que se hayan creado mediante el cifrado del lado del servidor con claves almacenadas en AWS KMS. En la configuración de CRR, ahora puede indicar a Amazon S3 que replique estos objetos. Para obtener más información, consulte Replicación de objetos cifrados (SSE-C, SSE-S3, SSE-KMS, DSSE-KMS).	06 de noviembre de 2017
Región de Europa (Londres)	Amazon S3 ya se encuentra disponible en la región UE (Londres) Para obtener más información acerca de las regiones y los puntos de conexión de Amazon S3, consulte Regiones y puntos de conexión en la Referencia general de AWS.	13 de diciembre de 2016
Región Canadá (Central)	Amazon S3 ya está disponible en la región Canadá (Central) Para obtener más información acerca de las regiones y los puntos de conexión de Amazon S3, consulte Regiones y puntos de conexión en la Referencia general de AWS.	8 de diciembre de 2016

Cambio	Descripción	Fecha
Etiquetado de objetos	Amazon S3 ya es compatible con el etiquetado de objetos. El etiquetado de objetos le permite categoriz ar el almacenamiento. Los prefijos de nombre de clave de objeto también le permiten categorizar el almacenam iento, pero el etiquetado de objetos agrega una dimensión adicional. El etiquetado agrega beneficios adicionales. Entre ellas se incluyen: • El etiquetado de objetos permite un control de acceso preciso de los permisos (por ejemplo: podría conceder un permiso de usuario de IAM a objetos de solo lectura con etiquetas específicas). • Control preciso en la especificación de una configura ción de ciclo de vida. Puede especificar etiquetas para seleccionar un subconjunto de objetos a los que se aplique la regla de ciclo de vida. • Si ha configurado la replicación en multirregiones (CRR), Amazon S3 puede replicar las etiquetas. Debe conceder los permisos necesarios al rol de IAM creado para que Amazon S3 asuma que debe replicar los objetos en su nombre. • También puede personalizar las métricas de CloudWatc h y los eventos de CloudTrail para mostrar información mediante filtros de etiquetas específicos. Para obtener más información, consulte Categorización del almacenamiento mediante etiquetas.	29 de noviembre de 2016

Cambio	Descripción	Fecha
El ciclo de vida de Amazon S3 ahora admite filtros basados en etiquetas	Amazon S3 ahora admite el filtrado basado en etiquetas en la configuración del ciclo de vida. Ahora puede especificar reglas del ciclo de vida en la que puede establecer un prefijo de clave, una o varias etiquetas de objeto o una combinación de ambos factores para seleccionar un subconjunto de objetos al que aplicar la regla del ciclo de vida. Para obtener más información, consulte Administración del ciclo de vida del almacenamiento.	29 de noviembre de 2016
Métricas de solicitudes de CloudWatch para buckets	Amazon S3 ahora admite métricas de CloudWatch para solicitudes realizadas en buckets. Cuando habilita estas métricas en un bucket, se informa de las métricas a intervalos de 1 minuto. También puede configurar qué objetos de un bucket informarán de estas métricas de solicitudes. Para obtener más información, consulte Monitorización de métricas con Amazon CloudWatch.	29 de noviembre de 2016
Inventario de Amazon S3	Ahora Amazon S3 es compatible con el inventario de almacenamiento. Amazon S3 Inventory proporciona una salida de archivos sin formato de los objetos y metadatos correspondientes diaria o semanalmente en un bucket de S3 o un prefijo compartido (es decir, objetos con nombres que comienzan con la misma cadena). Para obtener más información, consulte Inventario de Amazon S3.	29 de noviembre de 2016

Cambio	Descripción	Fecha
Análisis de Amazon S3: análisis de clases de almacenamiento	La nueva característica de análisis de las clases de almacenamiento de Amazon S3 observa los patrones de acceso a los datos para ayudarle a determinar cuándo trasladar el almacenamiento `STANDARD` al que se acceda con menos frecuencia a la clase de almacenamiento `STANDARD_IA` (IA, para acceso poco frecuente). Después de que el análisis de clase de almacenamiento observe estos patrones de acceso poco frecuentes de un conjunto de datos filtrados durante un periodo determinado de tiempo, puede usar los resultados del análisis para mejorar la configuración del ciclo de vida. Esta función también incluye un análisis diario detallado de su uso del almacenamiento en el bucket, prefijo o nivel de etiqueta especificado, que podrá exportar a un bucket de S3.	29 de noviembre de 2016
Nuevas recuperaciones de datos rápidas y en bloque al restaurar objetos archivados desde S3 Glacier	Amazon S3 ahora es compatible con las recuperaciones de datos rápidas y en bloque, además de las recuperaciones estándar, al restaurar objetos archivados a S3 Glacier. Para obtener más información, consulte Restauración de un objeto archivado.	21 de noviembre de 2016
Registro de objetos mediante CloudTrail	CloudTrail permite que se registren operaciones de API en el nivel de objetos de Amazon S3 como, por ejemplo, `GetObject` , `PutObject` y `DeleteObject` . Puede configurar los selectores de eventos para registrar operaciones de API de objeto. Para obtener más información, consulte Registro de llamadas a la API de Amazon S3 mediante AWS CloudTrail.	21 de noviembre de 2016
Región EE.UU Este (Ohio)	Amazon S3 ya está disponible en la región EE. UU. Este (Ohio). Para obtener más información acerca de las regiones y los puntos de conexión de Amazon S3, consulte Regiones y puntos de conexión en la Referencia general de AWS.	17 de octubre de 2016

Cambio	Descripción	Fecha
Compatibilidad de IPv6 con Amazon S3 Transfer Acceleration	Amazon S3 ahora admite el protocolo de Internet versión 6 (IPv6) para Amazon S3 Transfer Acceleration. Puede conectarse a Amazon S3 por IPv6 mediante la nueva doble pila para el punto de conexión de Transfer Acceleration. Para obtener más información, consulte Introducción a Amazon S3 Transfer Acceleration.	6 de octubre de 2016
Compatibilidad con IPv6	Amazon S3 ahora admite el protocolo de Internet versión 6 (IPv6). Puede obtener acceso a Amazon S3 por IPv6 utilizando puntos de enlace de doble pila. Para obtener más información, consulte Realización de solicitudes a Amazon S3 mediante IPv6.	11 de agosto de 2016
Región Asia Pacífico (Mumbai)	A partir de ahora, Amazon S3 está disponible en la región Asia-Pacífico (Mumbai). Para obtener más informaci ón acerca de las regiones y los puntos de conexión de Amazon S3, consulte Regiones y puntos de conexión en la Referencia general de AWS.	27 de junio de 2016
Amazon S3 Transfer Acceleration	Amazon S3 Transfer Acceleration facilita la transferencia rápida, sencilla y segura de archivos a largas distancia s entre su cliente y un bucket de Amazon S3. Transfer Acceleration aprovecha las ubicaciones de borde distribui das globalmente en Amazon CloudFront. Para obtener más información, consulte Configuración de transferencias de archivos rápidas y seguras con Amazon S3 Transfer Acceleration.	19 de abril de 2016
Compatibilidad del ciclo de vida para eliminar marcadore s de eliminación de objetos que vencieron	La acción `Expiration` de la configuración del ciclo de vida ahora permite indicarle a Amazon S3 que elimine los marcadores de eliminación de objetos que vencieron en un bucket con el control de versiones activado. Para obtener más información, consulte Elementos para describir las acciones del ciclo de vida.	16 de marzo de 2016

Cambio	Descripción	Fecha
La configuración del ciclo de vida del bucket admite acciones para detener cargas multiparte incompletas.	Ahora, la configuración del ciclo de vida de un bucket admite la acción `AbortIncompleteMultipartUpload`, que puede utilizar para pedirle a Amazon S3 que detenga las cargas multipartes que no se completan dentro de un periodo especificado de días después de iniciarse. Cuando una carga multiparte cumple los requisitos para una operación de detención, Amazon S3 elimina cualquier parte cargada y detiene la carga multiparte. Para obtener información conceptual, consulte los siguientes temas en la Guía del usuario de Amazon S3: • Anulación de la carga multiparte • Elementos para describir las acciones del ciclo de vida Las siguientes operaciones de la API se han actualizado para admitir la nueva acción: • PUT Bucket lifecycle: ahora, la configuración XML le permite especificar la acción `AbortIncompleteMultipartUpload` en una regla de configuración de ciclo de vida. • List Parts e Initiate Multipart Upload: estas dos operaciones de la API ahora devuelven dos encabezados de respuesta adicionales (`x-amz-abort-date` y `x-amz-abort-rule-id`) si el bucket tiene una regla del ciclo de vida que especifique la acción `AbortIncompleteMultipartUpload`. Estos encabezados de respuesta indican si la carga multiparte iniciada cumplirá los requisitos de la operación de detención y qué regla del ciclo de vida es aplicable.	16 de marzo de 2016

Cambio	Descripción	Fecha
Región Asia Pacífico (Seúl)	Amazon S3 ya está disponible en la región Asia-Pacífico (Seúl). Para obtener más información acerca de las regiones y los puntos de conexión de Amazon S3, consulte Regiones y puntos de conexión en la Referencia general de AWS.	6 de enero de 2016
Nueva clave de condición y cambio en la carga multiparte	Ahora, las políticas de IAM admiten una clave de condición `s3:x-amz-storage-class` de Amazon S3. Para obtener más información, consulte Ejemplos de políticas de bucket que utilizan claves de condición. Ya no tiene por qué ser el iniciador de una carga multiparte para cargar partes y completar la carga. Para obtener más información, consulte API y permisos de carga multiparte.	14 de diciembre de 2015
Cambio de nombre de la región EE.UU. Estándar	Se ha cambiado la cadena del nombre de la región de "EE.UU. Estándar" a "EE.UU. Este (Norte de Virginia)". Solo se ha cambiado el nombre de la región, no se cambia su funcionalidad.	11 de diciembre de 2015

Cambio	Descripción	Fecha
Nueva clase de almacenamiento	Amazon S3 ahora ofrece una nueva clase de almacenamiento, STANDARD_IA (IA quiere decir acceso poco frecuente) para el almacenamiento de objetos. Esta clase de almacenamiento está optimizada para los datos de duración prolongada y a los que se obtenga acceso con menor frecuencia. Para obtener más información, consulte Uso de las clases de almacenamiento de Amazon S3. Las actualizaciones de la función de configuración del ciclo de vida ahora le permiten realizar una transición de objetos a la clase de almacenamiento STANDARD_IA. Para obtener más información, consulte Administración del ciclo de vida del almacenamiento. Anteriormente, la función de replicación entre regiones usaba la clase de almacenamiento del objeto original para las réplicas de objetos. Ahora, cuando configure la replicación entre regiones puede especificar una clase de almacenamiento para la réplica del objeto creada en el bucket de destino. Para obtener más información, consulte Información general de la replicación de objetos.	16 de septiembre de 2015
AWS CloudTrail Integración de	La nueva integración de AWS CloudTrail le permite registrar la actividad de la API de Amazon S3 en su bucket de S3. Puede usar CloudTrail para realizar un seguimiento de las creaciones o eliminaciones de buckets de S3, modificaciones en el control de acceso o cambios en la configuración del ciclo de vida. Para obtener más información, consulte Registro de llamadas a la API de Amazon S3 mediante AWS CloudTrail.	1 de septiembre de 2015

Cambio	Descripción	Fecha
Aumento del límite de buckets	Amazon S3 ahora admite aumentos en los límites de buckets. De forma predeterminada, los clientes pueden crear hasta 100 buckets en su Cuenta de AWS. Los clientes que necesiten buckets adicionales pueden aumentar el límite solicitando un aumento en el límite de servicio. Para obtener información acerca de cómo aumentar su límite de buckets, consulte Cuotas de Servicio de AWS en la Referencia general de AWS. Para obtener más información, consulte Uso de la SDKs AWS y Cuotas, restricciones y limitaciones de bucket.	4 de agosto de 2015
Actualización del modelo de consistencia	Amazon S3 ahora es compatible con la coherencia de lectura tras escritura para objetos nuevos añadidos a Amazon S3 en la región EE. UU. Este (Norte de Virginia) . Antes de esta actualización, todas las regiones excepto EE. UU. Este (Norte de Virginia) eran compatibles con la coherencia de lectura tras escritura para objetos nuevos cargados en Amazon S3. Con esta mejora, Amazon S3 ahora es compatible con la coherencia de lectura tras escritura en todas las regiones para objetos nuevos añadidos a Amazon S3. La coherencia de lectura tras escritura le permite recuperar objetos inmediatamente tras su creación en Amazon S3. Para obtener más información, consulte Regiones.	4 de agosto de 2015
Notificaciones de eventos	Las notificaciones de eventos de Amazon S3 se han actualizado para agregar notificaciones cuando se eliminan los objetos y para agregar filtros por nombres de objeto con coincidencia por prefijo y sufijo. Para obtener más información, consulte Notificaciones de eventos de Amazon S3.	28 de julio de 2015

Cambio	Descripción	Fecha
Integración en Amazon CloudWatch	La nueva integración de Amazon CloudWatch le permite monitorear y establecer alarmas sobre el uso de Amazon S3 mediante métricas de CloudWatch para Amazon S3. Entre las métricas compatibles se incluyen el número total de bytes para el almacenamiento estándar, el número total de bytes para el almacenamiento de redundancia reducida y el número total de objetos para un bucket de S3 dado. Para obtener más información, consulte Monitorización de métricas con Amazon CloudWatch.	28 de julio de 2015
Compatibilidad para la eliminación y el vaciado de buckets no vacíos	Amazon S3 ahora permite la eliminación y el vaciado de buckets no vacíos. Para obtener más información, consulte Vaciar un bucket.	16 de julio de 2015
Políticas de bucket para puntos de conexión de Amazon VPC	Amazon S3 ha añadido compatibilidad con políticas de bucket para puntos de enlace de la nube virtual privada (VPC). Puede utilizar las políticas de bucket de S3 para controlar el acceso a los buckets desde puntos de enlace específicos de la VPC o VPC específicas. Los puntos de enlace de la VPC se configuran fácilmente, son muy fiables y ofrecen una conexión segura con Amazon S3 sin necesidad de utilizar una gateway ni una instancia NAT. Para obtener más información, consulte Control del acceso desde puntos de enlace de la VPC con políticas de bucket.	29 de abril de 2015
Notificaciones de eventos	Se han actualizado las notificaciones de eventos de Amazon S3 para admitir el cambio a permisos basados en recursos para las funciones de AWS Lambda. Para obtener más información, consulte Notificaciones de eventos de Amazon S3.	9 de abril de 2015

Cambio	Descripción	Fecha
Replicación entre regiones	Amazon S3 ahora permite la replicación entre diferente s regiones La reproducción entre regiones consiste en la copia automática y asíncrona de los objetos en los buckets de diferentes Regiones de AWS. Para obtener más información, consulte Información general de la replicación de objetos.	24 de marzo de 2015
Notificaciones de eventos	Amazon S3 ahora es compatible con nuevos tipos de eventos y destinos en una configuración de notificaciones de bucket. Antes de esta versión, Amazon S3 solo admitía el tipo de evento s3:ReducedRedundancyLostObject y un tema de Amazon SNS como destino. Para obtener más información sobre los nuevos tipos de evento, consulte Notificaciones de eventos de Amazon S3.	13 de noviembre de 2014
Cifrado en el servidor con claves de cifrado proporcionadas por el cliente	Cifrado del lado del servidor con claves AWS Key Management Service (AWS KMS) (SSE-KMS) Amazon S3 ahora admite el cifrado del lado del servidor con AWS KMS. Esta característica le permite administr ar la clave de sobre a través de AWS KMS, y Amazon S3 llama a AWS KMS para acceder a la clave de sobre con los permisos que establezca. Para obtener más información acerca del cifrado del lado del servidor con AWS KMS, consulte Protección de datos mediante cifrado del lado del servidor con AWS Key Management Service.	12 de noviembre de 2014
Región de Europa (Fráncfort)	Amazon S3 ya está disponible en la región de UE (Fráncfor t).	23 de octubre de 2014

Cambio	Descripción	Fecha
Cifrado en el servidor con claves de cifrado proporcionadas por el cliente	A partir de ahora, Amazon S3 admite el cifrado en el servidor con claves de cifrado proporcionadas por el cliente (SSE-C). El cifrado en el servidor le permite solicitar a Amazon S3 que cifre sus datos en reposo. Al usar SSE-C, Amazon S3 cifra sus objetos con las claves de cifrado personalizadas que facilite. Dado que Amazon S3 realiza el cifrado por usted, disfrutará de los beneficios de usar sus propias claves de cifrado sin el coste derivado de escribir o ejecutar su propio código de cifrado. Para obtener más información sobre SSE-C, consulte Cifrado del lado del servidor (con claves de cifrado proporcionadas por el cliente).	12 de junio de 2014
Compatibilidad de ciclo de vida para el control de versiones.	Antes de esta versión, la configuración del ciclo de vida solo se permitía en los buckets no habilitados para el control de versiones. Ahora puede configurar el ciclo de vida tanto en buckets sin control de versiones como en buckets habilitados para el control de versiones. Para obtener más información, consulte Administración del ciclo de vida del almacenamiento.	20 de mayo de 2014
Revisión de los temas de control de acceso	Se ha revisado la documentación sobre control de acceso de Amazon S3. Para obtener más información, consulte Administración de identidades y accesos para Amazon S3.	15 de abril de 2014
Se ha revisado el tema de registro de acceso al servidor	Se ha revisado la documentación de registro de acceso al servidor. Para obtener más información, consulte Registro de solicitudes con registro de acceso al servidor.	26 de noviembre de 2013
Actualización de las muestras del SDK de .NET a la versión 2.0	Las muestras del SDK de .NET de esta guía ahora con compatibles con la versión 2.0.	26 de noviembre de 2013

Cambio	Descripción	Fecha
Compatibilidad con SOAP por HTTP obsoleta	La compatibilidad con SOAP por HTTP está obsoleta, pero aún se encuentra disponible con HTTPS. Las nuevas características de Amazon S3 no serán compatibles con SOAP. Le recomendamos que utilice la API de REST o los SDK de AWS.	20 de septiembre de 2013
Compatibilidad con variables de políticas de IAM	El lenguaje de la política de IAM ya es compatible con variables. Cuando se evalúa una política, las variables de la política se sustituyen por valores facilitados por información basada en contexto desde la sesión del usuario autenticado. Puede utilizar las variables de políticas para definir políticas con un propósito general sin mostrar un listado explícito con todos los componentes de la política. Para obtener más información acerca de las variables de políticas, consulte Introducción a las variables de las políticas de IAM en la guía del usuario de IAM. Para obtener algunos ejemplos de variables de políticas de Amazon S3, consulte Ejemplos de políticas basadas en identidad para Amazon S3.	3 de abril de 2013
Compatibilidad de la consola con pagos por solicitante	Ahora puede configurar su bucket de pago por solicitante con la consola de Amazon S3. Para obtener más información, consulte Utilización de buckets de pago por solicitante para transferencias de almacenamiento y uso.	31 de diciembre de 2012

Cambio	Descripción	Fecha
Soporte del dominio raíz para alojamiento de sitios web	Amazon S3 ahora es compatible con el alojamiento de sitios web estáticos en el dominio raíz. Los visitantes de su sitio web pueden obtener acceso a su sitio desde su navegador sin especificar www en la dirección web (por ejemplo, pueden usar example.com en vez de www.example.com). Muchos clientes ya cuentan con alojamientos de sitios web estáticos en Amazon S3, a los que acceden a través un subdominio www (por ejemplo, www.example.com). Anteriormente, para permitir el acceso al dominio raíz, necesitaba ejecutar sus propias solicitudes desde el servidor web al dominio raíz del proxy desde navegadores a su sitio web en Amazon S3. La ejecución de solicitudes desde el servidor web al proxy introduce costos adicionales, presión operativa y un nuevo punto potencial de errores. Ahora, puede aprovechar un alto nivel de disponibilidad y durabilidad de Amazon S3 tanto para las direcciones www como para las del dominio raíz. Para obtener más información, consulte Alojamiento de un sitio web estático mediante Amazon S3.	27 de diciembre de 2012
Revisión de consola	Se ha actualizado la consola de Amazon S3. Los temas de la documentación que se refieren a la consola se han revisado correspondientemente.	14 de diciembre de 2012

Cambio	Descripción	Fecha
Compatibilidad para archivar datos en S3 Glacier	A partir de ahora, Amazon S3 admite una opción de almacenamiento que le permite usar el servicio de almacenamiento a bajo coste de S3 Glacier para el archivado de datos. Para archivar objetos, debe definir reglas de archivo para identificar los objetos y el periodo de tiempo en que desea que Amazon S3 archive dichos objetos en S3 Glacier. Puede establecer fácilmente las reglas en un bucket con la consola de Amazon S3 o mediante programación con la API de Amazon S3 o los SDK de AWS. Para obtener más información, consulte Administración del ciclo de vida del almacenamiento.	13 de noviembre de 2012
Soporte de redirecci onamiento de páginas web	Para los buckets configurados como sitios web, Amazon S3 permite ahora redirigir una solicitud desde un objeto a otro objeto del mismo bucket o a una URL externa. Para obtener más información, consulte (Opcional) Configura ción del redireccionamiento de páginas web. Para obtener más información acerca del alojamiento de sitios web, consulte Alojamiento de un sitio web estático mediante Amazon S3.	4 de octubre de 2012
Soporte de uso compartido de recursos entre orígenes (CORS)	A partir de ahora, Amazon S3 permite el uso compartid o de recursos entre orígenes (CORS). CORS define una forma en la que las aplicaciones web clientes cargadas en un dominio pueden interactuar u obtener acceso a los recursos de un dominio distinto. Con la compatibilidad de CORS en Amazon S3, puede desarrollar aplicaciones web completas del lado del cliente sobre Amazon S3 y permitir un acceso entre dominios de forma selectiva a sus recursos de Amazon S3. Para obtener más informaci ón, consulte Uso compartido de recursos entre orígenes (CORS).	31 de agosto de 2012

Cambio	Descripción	Fecha
Soporte de etiquetado de asignación de costos	A partir de ahora, Amazon S3 permite usar el etiquetado de asignación de costes, lo que le permite etiquetar los buckets de S3 de manera que pueda realizar un seguimiento de los costes según los proyectos y otros criterios más fácilmente. Para obtener más información acerca del etiquetado de buckets, consulte Uso de etiquetas de buckets de S3 de asignación de costos.	21 de agosto de 2012
Compatibilidad con el acceso a la API protegido por MFA en políticas de buckets	Amazon S3 ahora admite el acceso a la API protegido por MFA, una característica que permite emplear la autenticación multifactor de AWS para conseguir un nivel extra de seguridad en el momento de acceder a sus recursos de Amazon S3. Se trata de una característica de seguridad que requiere que los usuarios demuestren una posesión física de un dispositivo de MFA facilitando un código MFA válido. Para obtener más información, consulte Autenticación multifactor de AWS. Ahora puede solicitar la autenticación MFA para cualquier solicitud de acceso a sus recursos de Amazon S3. Para implementar la autenticación MFA, ahora Amazon S3 permite usar la clave `aws:MultiFactorAuthAge` en una política de bucket. Para ver una política de bucket de ejemplo, consulte Exigir MFA.	10 de julio de 2012
Compatibilidad con el vencimiento de objetos	Puede usar el vencimiento de objetos para programar la eliminación automática de datos tras un periodo de tiempo configurado. Para establecer el vencimiento de un objeto puede añadir una configuración del ciclo de vida a un bucket.	27 de diciembre de 2011
Compatibilidad con nueva región	Amazon S3 ahora es compatible con la región de Sudamérica (São Paulo). Para obtener más información, consulte Acceso y publicación de un bucket de Amazon S3.	14 de diciembre de 2011

Cambio	Descripción	Fecha
Eliminar varios objetos	A partir de ahora, Amazon S3 es compatible con la API de eliminación de varios objetos, que le permite eliminar varios objetos en una sola solicitud. Con esta caracterí stica, podrá eliminar grandes cantidades de objetos de Amazon S3 con mayor rapidez que usando varias solicitud es DELETE individuales. Para obtener más información, consulte Eliminación de objetos de Amazon S3.	7 de diciembre de 2011
Compatibilidad con nueva región	Amazon S3 admite ahora la región EE. UU. Oeste (Oregón). Para obtener más información, consulte Buckets y regiones.	8 de noviembre de 2011
Actualización de documentación	Correcciones de errores en la documentación.	8 de noviembre de 2011
Actualización de documentación	Además de las correcciones de errores en la documenta ción, esta versión incluye las siguientes mejoras: • Nuevas secciones de cifrado del lado del servidor con AWS SDK for PHP y AWS SDK for Ruby (consulte Especificación del cifrado del servidor con claves administradas por Amazon S3 (SSE-S3)).	17 de octubre de 2011
Compatibilidad con el cifrado en el servidor	A partir de ahora, Amazon S3 es compatible con el cifrado en el lado del servidor. Le permite solicitar a Amazon S3 que cifre sus datos en reposo, es decir, que cifre sus datos de objetos cuando Amazon S3 los escribe en discos en sus centros de datos. Además de las actualizaciones de la API de REST, AWS SDK for Java y .NET proporcionan la funcionalidad necesaria para solicitar el cifrado en el servidor. También puede solicitar el cifrado del lado del servidor cuando carga objetos con la AWS Management Console. Para obtener más información sobre el cifrado de datos, consulte Protección de datos mediante cifrado.	4 de octubre de 2011

Cambio	Descripción	Fecha
Actualización de documentación	Además de las correcciones de errores en la documentación, esta versión incluye las siguientes mejoras: • Nuevas muestras de Ruby y PHP en la sección Realizar solicitudes. • Se han agregado secciones que describen cómo generar y usar las URL prefirmadas. Para obtener más información, consulte Uso compartido de objetos con URL prefirmadas y Uso compartido de objetos con URL prefirmadas. • Se ha actualizado una sección existente para introducir los exploradores de AWS para Eclipse y Visual Studio. Para obtener más información, consulte Desarrollo con Amazon S3 mediante los SDK de AWS.	22 de septiembre de 2011

Cambio	Descripción	Fecha
Compatibilidad para enviar solicitudes con credenciales de seguridad temporales	Además de usar las credenciales de seguridad de su Cuenta de AWS y de su usuario de IAM para enviar solicitudes autenticadas a Amazon S3, ahora puede enviar solicitudes con las credenciales de seguridad temporales que obtenga de AWS Identity and Access Management (IAM). Puede usar las bibliotecas de encapsulamiento de la API de AWS Security Token Service o de los SDK de AWS para solicitar estas credenciales temporales de IAM. Puede solicitar estas credenciales de seguridad temporales para uso propio o puede entregarlas a usuarios federados o aplicaciones. Esta característica le permite administrar a los usuarios fuera de AWS y proporcionarles credenciales temporales de seguridad para acceder a los recursos de AWS. Para obtener más información, consulte Realizar solicitudes. Para obtener más información acerca de la compatibilidad de IAM con las credenciales de seguridad temporales, consulte Credenciales de seguridad temporales en la guía de usuario de IAM.	3 de agosto de 2011
Ampliación de la API de carga multiparte para habilitar la copia de objetos de hasta 5 TB	Antes de esta versión, la API de Amazon S3 permitía la copia de objetos con un tamaño de hasta 5 GB. Para permitir la copia de objetos de más de 5 GB, ahora Amazon S3 amplía la API de carga multiparte con una nueva operación, `Upload Part (Copy)` . Puede usar esta operación de carga multiparte para copiar objetos con un tamaño de hasta 5 TB. Para obtener más información, consulte Copia, traslado y cambio de nombre de objetos. Para obtener información conceptual sobre la API de carga multiparte, consulte Carga y copia de objetos con la carga multiparte.	21 de junio de 2011

Cambio	Descripción	Fecha
Desactivación de las llamadas a la API de SOAP por HTTP	Para aumentar la seguridad, se han desactivado las llamadas a la API de SOAP por HTTP. Las solicitud es autenticadas y anónimas a SOAP deben enviarse a Amazon S3 con SSL.	6 de junio de 2011
IAM admite la delegación entre cuentas	Anteriormente, para obtener acceso a un recurso de Amazon S3, un usuario de IAM necesitaba permisos tanto de la Cuenta de AWS principal como del propietario del recurso de Amazon S3. Con el acceso entre cuentas, el usuario de IAM ahora solo necesita permiso del propietar io de la cuenta. Es decir, si el propietario de un recurso concede acceso a una Cuenta de AWS, la Cuenta de AWS puede ahora conceder acceso a estos recursos a los usuarios de IAM. Para obtener más información, vea Crear un rol para delegar permisos a un usuario de IAM en Guía del usuario de IAM. Para obtener más información sobre cómo especificar principales en una política de bucket, consulte Entidades principales de las políticas de bucket.	6 de junio de 2011
Nuevo enlace	La información de punto de conexión de este servicio se encuentra ahora en la Referencia general de AWS. Para obtener más información, vaya a Regiones y puntos de conexión en la Referencia general de AWS.	1 de marzo de 2011

Cambio	Descripción	Fecha
Compatibilidad con el alojamiento de sitios web estáticos en Amazon S3	Amazon S3 presenta una compatibilidad mejorada para albergar sitios web estáticos. Esto incluye soporte para documentos de índice y documentos de error personalizados. Al utilizar estas características, las solicitudes a la raíz de su bucket o una subcarpeta (por ejemplo, `http://mywebsite.com/subfolder`) devuelven el documento de índice en vez de la lista de objetos en su bucket. Si se encuentra un error, Amazon S3 devuelve su mensaje de error personalizado, en lugar de un mensaje de error de Amazon S3. Para obtener más información, consulte Alojamiento de un sitio web estático mediante Amazon S3.	6 de junio de 2011
La información de punto de conexión de este servicio se encuentra ahora en la Referencia general de AWS. Para obtener más información, vaya a Regiones y puntos de conexión en la Referencia general de AWS.	1 de marzo de 2011	

Cambio	Descripción	Fecha
Compatibilidad con el alojamiento de sitios web estáticos en Amazon S3	Amazon S3 presenta una compatibilidad mejorada para albergar sitios web estáticos. Esto incluye soporte para documentos de índice y documentos de error personalizados. Al utilizar estas características, las solicitudes a la raíz de su bucket o una subcarpeta (por ejemplo, `http://mywebsite.com/subfolder`) devuelven el documento de índice en vez de la lista de objetos en su bucket. Si se encuentra un error, Amazon S3 devuelve su mensaje de error personalizado, en lugar de un mensaje de error de Amazon S3. Para obtener más información, consulte Alojamiento de un sitio web estático mediante Amazon S3.	17 de febrero de 2011
Compatibilidad con encabezados de respuesta en la API	La API de REST de GET Object ahora permite cambiar los encabezados de respuesta de la solicitud GET Object de REST en cada caso. Es decir, puede alterar los metadatos del objeto en la respuesta sin alterar el objeto en sí. Para obtener más información, consulte Descarga de objetos.	14 de enero de 2011
Compatibilidad con objetos grandes	Amazon S3 ha incrementado el tamaño máximo de objetos que puede almacenar en un bucket de S3, de 5 GB a 5 TB. Si usa la API de REST, podrá cargar objetos de hasta 5 GB en una única operación PUT. Para objetos más grandes, debe usar la API de REST de carga multiparte para cargar objetos en partes. Para obtener más información, consulte Carga y copia de objetos con la carga multiparte.	9 de diciembre de 2010
Carga multiparte	La carga multiparte permite cargas más rápidas y flexibles en Amazon S3. Permite cargar un solo objeto como un conjunto de partes. Para obtener más información, consulte Carga y copia de objetos con la carga multiparte.	10 de noviembre de 2010

Cambio	Descripción	Fecha
Compatibilidad con ID canónicos en políticas de bucket	Ahora puede especificar ID canónicos en políticas de bucket. Para obtener más información, consulte Entidades principales de las políticas de bucket	17 de septiembre de 2010
Amazon S3 funciona con IAM	Este servicio ahora se integra con AWS Identity and Access Management (IAM). Para obtener más información, diríjase a Servicios de AWS que funcionan con IAM en la Guía del usuario de IAM.	2 de septiembre de 2010
Notificaciones	La característica de notificaciones de Amazon S3 le permite configurar un bucket de manera que Amazon S3 publique un mensaje en un tema de Amazon Simple Notification Service (Amazon SNS) cuando detecte un evento clave en un bucket. Para obtener más información, consulte Configuración de notificaciones de eventos de bucket.	14 de julio de 2010
Políticas de buckets	Las políticas de buckets conforman un sistema de administración de acceso que se usa para establecer permisos de acceso entre buckets, objetos y conjuntos de objetos. Esta funcionalidad suplementa, y en muchos casos sustituye, a las listas de control de acceso. Para obtener más información, consulte Políticas de buckets para Amazon S3.	6 de julio de 2010
Sintaxis estilo ruta disponibles en todas las regiones	Ahora, Amazon S3 admite la sintaxis estilo ruta para cualquier bucket en la región clásica de EE. UU., o si el bucket está en la misma región que el punto de conexión de la solicitud. Para obtener más información, consulte Alojamiento virtual.	9 de junio de 2010
Nuevo punto de conexión para UE (Irlanda)	Amazon S3 ofrece ahora un punto de conexión para UE (Irlanda): `http://s3-eu-west-1.amazonaws.com` .	9 de junio de 2010

Cambio	Descripción	Fecha
Consola	Ahora puede utilizar Amazon S3 a través de AWS Management Console. Lea sobre todas las funciones de Amazon S3 en la consola en la Guía del usuario de Amazon Simple Storage Service.	9 de junio de 2010
Redundancia reducida	Ahora, Amazon S3 le permite reducir sus costes de almacenamiento almacenando objetos en Amazon S3 con redundancia reducida. Para obtener más información, consulte Almacenamiento de redundancia reducida.	12 de mayo de 2010
Compatibilidad con nueva región	Amazon S3 ahora es compatible con la región de Asia-Pacífico (Singapur). Para obtener más información, consulte Buckets y regiones.	28 de abril de 2010
Control de versiones de objetos	Esta versión introduce el control de versiones de objetos. Ahora, todos los objetos pueden tener una clave y una versión. Si activa el control de versiones en un bucket, Amazon S3 da a todos los objetos que se añaden a un bucket un ID de versión exclusivo. Esta función le permite recuperarse de sobrescrituras y eliminaciones no intencion adas. Para obtener más información, consulte Control de versiones y Uso del control de versiones.	8 de febrero de 2010
Compatibilidad con nueva región	Amazon S3 ahora es compatible con la región EE. UU. Oeste (Norte de California). El nuevo punto de conexión para solicitudes en esta región es `s3-us-wes t-1.amazonaws.com` . Para obtener más información, consulte Buckets y regiones.	2 de diciembre de 2009

Cambio	Descripción	Fecha
AWS SDK for .NET	AWS ahora proporciona bibliotecas, código de muestra, tutoriales y otros recursos para los desarrolladores de software que prefieren crear aplicaciones con operaciones de la API específicas del lenguaje .NET en lugar de REST o SOAP. Estas bibliotecas proporcionan funciones básicas (que no se incluyen en las API de REST o SOAP), como la autenticación de solicitudes, los reintentos de solicitudes y la administración de errores para que se pueda comenzar más fácilmente. Para obtener más información sobre las bibliotecas y recursos específicos a lenguajes, consulte Desarrollo con Amazon S3 mediante los SDK de AWS.	11 de noviembre de 2009

Glosario de AWS

Para ver la terminología más reciente de AWS, consulte el Glosario de AWS en la Referencia de Glosario de AWS.